



HAL
open science

Conception et développement de robots durables

Salah Eddine S. E. Zerkane

► **To cite this version:**

Salah Eddine S. E. Zerkane. Conception et développement de robots durables. Robotique [cs.RO]. 2014. dumas-01088846

HAL Id: dumas-01088846

<https://dumas.ccsd.cnrs.fr/dumas-01088846>

Submitted on 28 Nov 2014

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

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



Distributed under a Creative Commons Attribution 4.0 International License



RESEARCH MASTER THESIS



RESEARCH MASTER THESIS

Conception et développement de robots durables

Auteur :
Salaheddine Zerkane

Encadrants :
Jean Vareille
Philippe Le Parc
Yvon Autret

SFIIS (Sécurité, Fiabilité et
Intégrité de l'Information et
des Systèmes)

UMR 6285 LAB-STICC



Résumé : La robotique durable s'intéresse à la conception et au développement de robots durables qui respectent l'environnement. Cette discipline assume ses premiers pas dans un monde inondé par l'augmentation de la consommation des ressources environnementales et des déchets vers un environnement affaibli par l'industrialisation de masse de notre époque. De ce fait, la robotique durable se présente comme une solution pour faire face à ces enjeux en robotique.

Dans cette perspective, ce rapport présente le méta modèle SuRo (*Sustainable Robotics*) pour la robotique durable. SuRo est une solution d'ingénierie dirigée par les modèles, qui a été conçue en intégrant les principes de la robotique, du génie logiciel et du développement durable. Il présente une première contribution dans ce domaine et une solution potentielle qui permet aux décideurs de faire des simulations de scénarios afin de choisir les configurations les plus durables. Le méta modèle SuRo a été proposé en se basant sur une analyse du domaine qui nous a permis d'étudier, de spécifier et de rassembler tous ses concepts. Pour son développement nous avons utilisé Eclipse Ecore qui est basé sur les notions de la MDA et qui permet de générer des squelettes de code en Java.

Mots clefs: Robotique, Génie Logicielle, Ingénierie dirigée par les modèles, Développement durable, MDA.

Abstract: Sustainable robotics emphasizes on designing and developing sustainable environmentally friendly robots. This branch of knowledge assumes its first steps into a world, flooded by the increase of environmental resources consumption and waste in an environment, already declining, because of the actual mass industrialization. Thus, sustainable robotics comes as a solution to deal with these issues in robotics.

In this scope, this report presents the meta-model SuRo for sustainable robotics. SuRo is a based model driven engineering solution, which has been designed by incorporating the principles of robotics, software engineering and sustainable development. It is a first endeavour in this field and a potential answer for decision makers to perform simulations of scenarios, in order to choose the most sustainable configurations. The meta-model SuRo was proposed based on an analysis of the field. This has allowed us to study, to specify and to assemble all its concepts. For its development we used Eclipse Ecore, a framework based on the concepts of MDA to generate Java skeleton code.

Keywords: Robotics, Software Engineering, Model Driven Engineering, Sustainable Development, MDA.

Sommaire

Chapitre 1 : Introduction

1.1.	Définition.....	5
1.2.	Evolution de la robotique.....	6
1.3.	Domaines de la robotique.....	6
1.4.	Motivation.....	8
1.5.	Hypothèses de travail.....	9
1.6.	Synthèse du chapitre 1.....	10

Chapitre 2 : Contexte de la robotique durable

2.1.	Etat de l'art de la robotique durable.....	11
2.2.	Enquête sur la robotique durable en France.....	12
2.3.	Synthèse du chapitre 2.....	13

Chapitre 3 : Analyse du domaine de la robotique durable

3.1.	Architecture des robots.....	14
3.2.	La partie matérielle.....	15
3.2.1.	Le châssis.....	15
3.2.2.	La structure mécanique.....	15
3.2.3.	Les actionneurs.....	17
3.2.4.	Les capteurs.....	17
3.2.5.	Le contrôleur.....	19
3.2.6.	Vers une structure matérielle durable.....	19
3.3.	La partie logicielle.....	22
3.3.1.	Les architectures logicielles en robotique.....	23
3.3.2.	Les composants logiciels.....	25
3.3.3.	Le logiciel durable.....	26
3.4.	Synthèse du chapitre 3.....	28

Chapitre 4 : Le méta modèle de la robotique durable

4.1.	L'ingénierie dirigée par les modèles.....	29
4.2.	Le CIM de SuRo.....	32
4.3.	Le PIM de SuRo.....	36
4.4.	Le PSM de SuRo.....	40
4.5.	Liens avec la méthode en X.....	40
4.6.	Synthèse du chapitre 4.....	411

Chapitre 5 : Conclusion	42
--------------------------------------	----

Références	44
-------------------------	----

Annexe 1 : Formulaire de l'enquête sur la robotique durable

Annexe 2 : ACV d'un robot compagnon

LISTE DES FIGURES

FIGURE 1 : CHRONOLOGIE DE LA ROBOTIQUE.....	7
FIGURE 2 : LE MODELE PROPOSE PAR L'INSTITUT FRAUNHOFER.....	12
FIGURE 3 : ARCHITECTURE GENERALE D'UN ROBOT.....	14
FIGURE 4 : CORPS MECANIQUES D'UN ROBOT.....	16
FIGURE 5: ARTICULATIONS D'UN BRAS ROBOTISE.....	16
FIGURE 6 : RELATION ENTRE LES ACTIONNEURS ET LES CORPS MECANIQUES.....	17
FIGURE 7 : PRINCIPE GENERAL DE FONCTIONNEMENT D'UN CAPTEUR.....	18
FIGURE 8: CYCLE DE VIE DU MATERIEL DU ROBOT.....	20
FIGURE 9: CATEGORIES D'IMPACTS ENVIRONNEMENTAUX.....	22
FIGURE 10 : LE MODELE D'ARCHITECTURE HIERARCHISEE.....	24
FIGURE 11 : LE MODELE D'ARCHITECTURE REACTIVE.....	24
FIGURE 12 : TAXONOMIE DES COMPOSANTS.....	26
FIGURE 13 : CYCLE DE VIE DU LOGICIEL ET SES IMPACTS.....	27
FIGURE 14 : RELATIONS ENTRE SYSTEME, MODELE ET META MODELE.....	29
FIGURE 15 : L'ARCHITECTURE MDA D'OMG.....	30
FIGURE 16 : TRANSFORMATION ET EXEMPLE DE MODELES EN MDA.....	31
FIGURE 17 : TRANSFORMATIONS MDA EN ROBOTIQUE.....	32
FIGURE 18 : IDEE PRINCIPALE DE SURO.....	33
FIGURE 19 : LE CIM DE SURO.....	35
FIGURE 20 : LE DIAGRAMME DE CLASSE DU META-META MODELE D'ECORE.....	37
FIGURE 21 : LE PIM DE BASE DE SURO.....	38
FIGURE 22 : LE NIVEAU 2 DE SURO.....	39
FIGURE 23 : LE NIVEAU 3 DE SURO.....	40
FIGURE 23 : LE SCHEMA DE LA METHODE EN X.....	41

LISTE DES TABLES

TABLEAU 1 : LES PRINCIPALES ENTREPRISES DE ROBOTIQUE EN FRANCE.....	13
TABLEAU 2 : LES TYPES DE METRIQUE DE DURABILITE DU LOGICIELLE.....	28
TABLEAU 3 : LA DEFINITION DES CONCEPTS DE SURO.....	34

*« Il ne sert de rien à l'homme de gagner
la lune s'il vient à perdre la Terre »*
François Mauriac (1885-1970)

Chapitre 1

Introduction

L'humanité est à l'aube d'une nouvelle ère, où le savoir constitue le centre de toutes les préoccupations et est une source majeure pour générer des biens, des services, de la richesse et pour produire de nouvelles connaissances. Cette vague de savoir s'implantera sur une vague industrielle mature, qui est caractérisée par une production de masse couplée à une consommation démesurée. Comment peut-on éviter que cela se traduise par un magma de pollution à l'échelle mondiale, et une décadence de l'environnement?

Dans ce paysage, l'être humain continue à exprimer sa motivation d'accroître ses capacités physiques et mentales pour explorer de nouveaux horizons, pour produire plus de biens et de services et pour vivre en prospérité. Les robots sont la conséquence de cette ambition inhérente, plus efficaces, plus performants, plus précis et maintenant, moins chers que la main-d'œuvre humaine, ils ont permis d'accroître la production de biens et de services quantitativement et qualitativement, et contribuent à notre développement dans plusieurs domaines et à plusieurs échelles. Qu'ils s'agissent de la complexité et de la diversité des tâches qu'ils peuvent effectuer, ou bien de leurs applications et de leurs frontières, ils ont admirablement surpassé celles de l'être humain et ont réussi à s'intégrer dans plusieurs aspects de sa vie.

La robotique connaît un engouement des plus spectaculaires, tant en industrie que dans la communauté scientifique. Son marché est en plein boom et ses domaines d'utilisation gagnent plus d'espace et plus de publique. La demande de robots augmentera encore plus dans les prochaines années. A titre indicatif, l'investissement initial dans les robots industriels est recouvert en 2 ans car le coût horaire d'un robot se limite essentiellement au coût horaire de son énergie consommée (typiquement 0.3 \$US) conduisant à un coût annuel 50 fois moins cher que celui d'une main-d'œuvre manuelle dans les pays développés [1]. En parallèle, le ratio de la dépendance des gens âgés et handicapés en Europe augmentera dans les années à venir, en 2060, 30 % des gens aurons plus de 65 ans et 12 % des personnes auront plus de 80 ans, ces individus dépendants aurons besoin de robots compagnons dans leurs maisons pour se maintenir et rester socialement connectés au monde [2]. Cette augmentation de ratio touchera aussi d'autre grand pays comme la Chine, le Japon et les USA.

Le robot est construit à partir de matières premières issues de l'environnement et à la fin de sa vie, il est mis au rebut dans cet environnement. Les phases des cycles de vie, de sa partie matérielle et de sa partie logicielle, échangent des flux énergétiques et de matières avec l'environnement, par la consommation de matières premières, d'énergie, ou bien, par le rejet de déchets. Les impacts qu'elles génèrent affectent l'environnement, à plusieurs niveaux et peuvent être mesurés par des indicateurs. Face à ces enjeux environnementaux, il devient essentiel de développer des méthodes et des modèles qui permettent de réduire les consommations de ressources, de minimiser les rejets dans l'environnement et améliorer les performances environnementales des robots.

Notre travail de recherche s'inscrit dans cette optique. Le méta modèle que nous proposons dans ce rapport permet d'apporter une solution à ces enjeux. Les concepts de robotique sont liés à ceux du génie logiciel et à ceux du développement durable pour modéliser un système de robotique durable. L'approche que nous proposons, permet d'analyser le domaine de robotique durable et de

construire un atelier logiciel qui pourra modéliser ce domaine afin de concevoir et développer des robots durables.

Ce rapport est organisé comme suit : ce premier chapitre introduit notre travail, la notion de robotique, son évolution, ses domaines, et décrit notre motivation et les hypothèses sur lesquelles se fondent notre travail. Le chapitre suivant, expose le contexte actuel de la robotique durable, nous présenterons les travaux dans ce domaine et les résultats d'une enquête que nous avons accomplie pour faire le point sur la situation en France. Dans le troisième chapitre, nous effectuons une analyse du domaine de la robotique, aux niveaux matériel et logiciel, tout en se penchant sur les liens avec le développement durable. Le quatrième chapitre décrit notre méta modèle créé selon l'analyse présentée dans le chapitre précédant, l'accent sera mis aussi sur sa relation avec la méthode en X. Et enfin, nous terminerons notre travail par une conclusion générale où nous afficherons les perspectives et ouvrirons la voix vers d'autres recherches.

1.1. Définition

Il existe dans la littérature plusieurs définitions du robot, par exemple l'organisation internationale de standardisation, dans le document ISO 8373 :2012, lui donne la définition d'un mécanisme programmable actionné sur au moins deux axes, avec un degré d'autonomie, se déplaçant dans son environnement, pour exécuter des tâches prévues ». Ou l'exemple de l'Institut Américain du Robot, qui le désigne comme un manipulateur reprogrammable multifonctionnel conçu pour déplacer du matériel, des pièces, des outils ou des dispositifs spéciaux par le biais de divers mouvements programmés pour l'exécution d'une variété de tâches. Ces deux définitions sont exclusives, elles concernent plutôt les robots industriels, les autres types de robots comme les humanoïdes, les robots de service ... etc. ne sont pas pris en compte. Elles sont aussi incomplètes, les aspects d'intelligence artificielle des robots, de perception et de prise de décision sont ignorés. On retrouve cette ambiguïté presque partout, par exemple le professeur Rodney Allen Brooks (ex-directeur du laboratoire d'informatique et d'intelligence artificielle à l'institut technologique du Massachusetts) décrit le robot comme suit : « Un robot est une machine qui détecte le monde, calcule, puis décide d'une action dans le monde qui a une portée physique au-delà de lui-même ». Dans la même perspective, cette définition n'inclut pas les robots virtuels (bot informatique) car elle se positionne dans le monde physique, en plus, elle ignore les aspects cognitifs du robot tels que la communication, l'apprentissage et la collaboration, finalement, le robot dont la partie 'calcul' est dans un *cloud* et dont les actions à exécuter peuvent lui être communiquées, n'est pas concerné par cette définition. Tout cela, reflète la complexité de la question de la frontière qui permet de cadrer la définition du robot, ce cadre reste toujours ouvert et vaste, Brooks implicitement l'aborde ; il ajoute à la fin de sa définition qu'il est possible de poursuivre sur le chemin de sa définition, ou bien sur d'autres, jusqu'à ce que celle-ci devienne obsolète.

En se basant sur ces définitions, nous présentons la nôtre de la manière suivante : le robot est une entité créée ; composée d'une partie logicielle incorporée dans une structure matérielle, qui imite les comportements observés dans le monde en utilisant des technologies issus de la mécanique, de l'informatique, de l'électronique et de l'intelligence artificielle, pour interagir avec son environnement. Notre définition est générique car elle concerne n'importe quel type de robot, elle reflète aussi le fondement de notre pensée. En effet, nous considérons le monde composé d'éléments immatériels (ex : information, etc.) incorporés dans la matière. Quelques soient les moyens d'interpréter cette complexité. L'information a toujours besoin d'une structure matérielle pour être traitée et stockée, de même, une structure matérielle à n'importe quel niveau a besoin de l'information pour être utilisable. Donc, nous considérons le robot comme une entité qui abstrait notre vision du monde par l'incorporation d'une partie immatérielle (logicielle) dans le matériel.

Nous avons aussi introduit la notion d'interaction avec l'environnement ; le robot est toujours associé à un environnement avec lequel il échange des flux.

1.2. Evolution de la robotique

Depuis les temps anciens, l'imagination de l'être humain n'a cessé de concevoir l'idée d'humanoïdes compagnons. Notre rêve de créer des artefacts capables de nous imiter et de nous aider dans notre quotidien, a toujours fait partie de notre histoire. La figure 1 illustre la chronologie de la robotique, avec les découvertes et les inventions scientifiques importantes qui l'ont marquée. La robotique a évolué durant les générations suivantes :

- a) *Génération des systèmes mécaniques purs* : Les robots de cette génération étaient des automates mécaniques, utilisant des techniques de pneumatique ou d'hydraulique pour effectuer des actions répétitives. Plusieurs inventions ont été élaborées comme : les humanoïdes musiciens d'al-Jazari capables de faire plus de 50 expressions corporelles, les automates de Jacques de Vaucanson (le canard mécanique, le joueur de flute), etc.
- b) *Génération des systèmes électriques* : Grâce à la découverte de l'électricité, de la radio et l'invention des moteurs à courant alternatif et continu, plusieurs parties mécaniques ont été remplacées par des composants électriques, permettant des meilleures performances. Un exemple est le « Teleautomaton » de Nikola Tesla ; un navire contrôlé à distance [3].
- c) *Génération des systèmes de contrôle automatique* : Grâce à l'utilisation des commutateurs, des solénoïdes, des transistors, etc. les comportements des composants mécaniques des robots ont pu être régulés, dirigés et commandés automatiquement. Exemple : Le robot « Elektro » de la Westinghouse Electric Corporation, qui marchait par des commandes vocales et pouvait bouger la tête et les bras [4].
- d) *Génération des systèmes électroniques* : dans cette période les progrès en informatique et en électronique comme les processeurs, circuits intégrés, langage de programmation Fortran, les systèmes d'exploitation ... etc. permirent de faire d'énormes avancées dans et l'automatisation des procédés industriels et le control électronique. En conséquence, ils conduisirent à la miniaturisation des composants du robot et à l'amélioration de la précision et de la rapidité des calculs. Exemple : Le robot industriel « ASEA IRB », qui a été le premier robot commandé par microprocesseur dans le monde. Il a été utilisé pour la manutention, l'emballage, le transport, le polissage, la soudure, et le classement [5].
- e) *Génération des systèmes intégrés* : L'invention des microcontrôleurs et les progrès en informatique permirent l'intégration du logiciel dans les composants matériels des robots. L'intelligence artificielle quant à elle, a permis de doter les robots de capacités cognitives qui leurs permettent généralement d'observer leurs environnements (Perception), de raisonner sur les données collectées (Raisonnement), de prendre des décisions selon les résultats du raisonnement et les connaissances acquises (Décision) et enfin d'exécuter les tâches correspondantes (Action). Avec l'évolution du génie logiciel, des télécommunications et des avancées dans les domaines de l'acquisition des données, des nanotechnologies, etc. ces robots ont investi plusieurs aspects de notre vie et sont devenus indispensables. Ont vu le jour des robots autonomes, des robots humanoïdes qui imitent les expressions et les mouvements corporels humains, etc.

1.3. Domaines de la robotique

Les robots ont réussi à investir plusieurs sphères de la vie humaine. Principalement, grâce à leurs performances dans la réalisation de leurs missions. L'utilisation des robots se répand à grande vitesse parmi plusieurs domaines, citons :

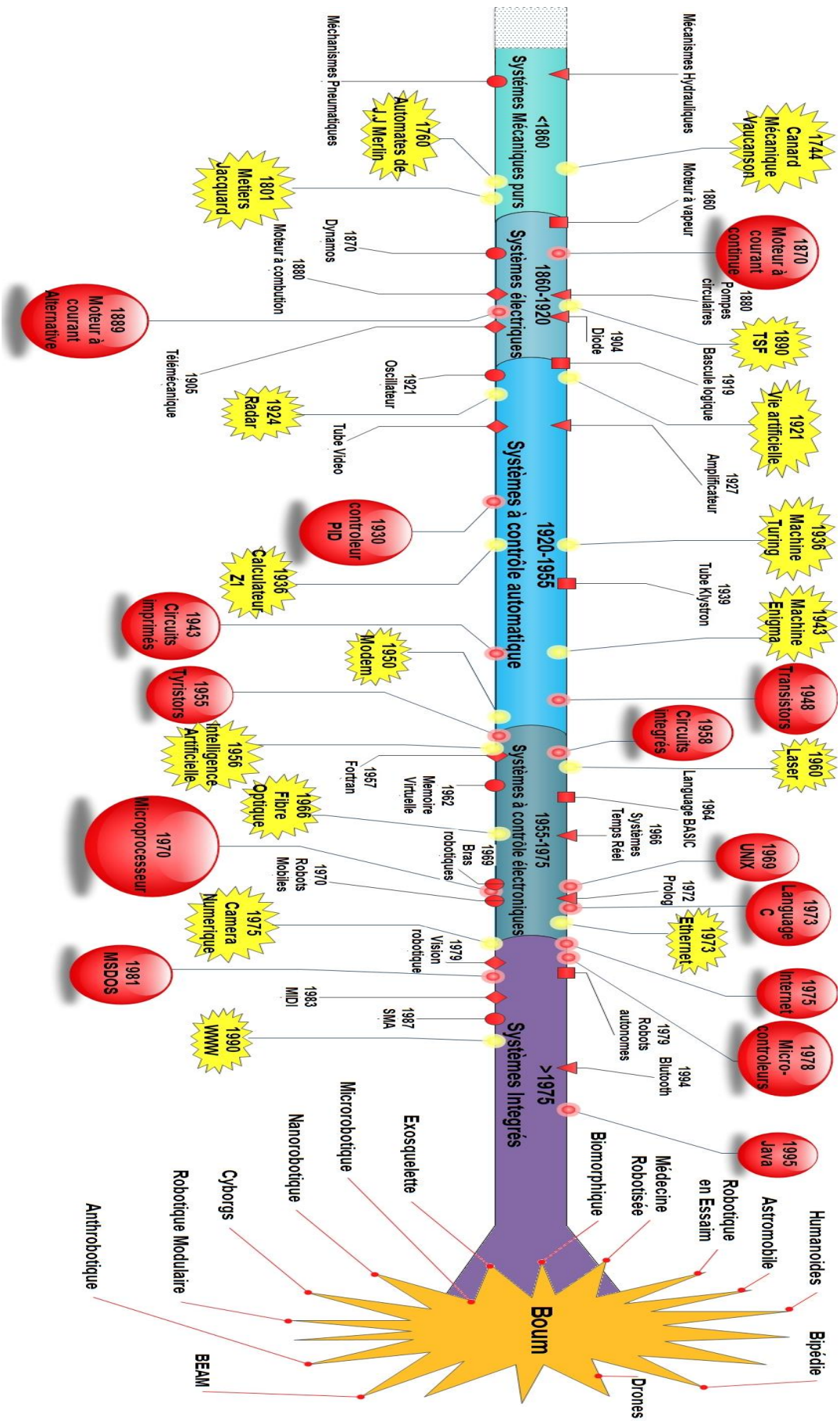


Figure 1 : Chronologie de la robotique.

- *En industrie* : Les robots ont été introduits en industrie dès les années 60. Leurs avantages ont été considérables, car ils ont permis d'augmenter la productivité avec des coûts réduits et surpassant ceux de la main d'œuvre humaine. Ils ont réussi également, à accomplir avec efficacité des tâches pénibles et complexes. Généralement, les robots industriels sont utilisés dans les chaînes d'assemblage, de fabrication, de transport, etc. la manutention, l'inspection, les tests, La maintenance industrielle, etc.

- *En médecine* : la précision et l'insensibilité aux facteurs naturels auxquels le chirurgien humain peut être soumis (comme le tremblement, la transpiration, le stress ou le tousotement), sont des caractéristiques des robots assistant en chirurgie. Ils sont employés à plusieurs niveaux en chirurgie pour aider le chirurgien à faire un travail plus efficace et avec moins de risque. Dans les hôpitaux les robots ont été employés comme aides-soignants pouvant interagir avec les malades et leur administrer leurs médicaments. La robotique a permis de développer des prothèses robotisées reproduisant la mécanique des membres humains, plusieurs personnes amputées ont pu retrouver certaines de leurs activités physiques grâce à ces robots. En outre, les robots sont utilisés en thérapie robotique ; ces robots contrôlés par un thérapeute permettent au système nerveux d'une personne handicapée de se réhabiliter, remplaçant la thérapie classique qui nécessitait un grand effort physique et plusieurs thérapeutes par patient. Enfin en industrie pharmaceutique, les robots sont utilisés pour la préparation et la fabrication automatisée des médicaments.

- *Dans l'exploration*: Les robots ont permis à l'être-humain d'explorer des environnements dangereux ou des lieux inaccessibles comme l'espace, les volcans, les fonds marins, etc. On envoie généralement ces robots pour collecter des informations environnementales, des images, des échantillons ou bien rechercher des personnes ou des objets dans le cas de désastre ou d'accident.

- *Dans le militaire et la sécurité* : Les robots sont utilisés par la police pour accéder à des lieux risqués afin de collecter des informations, ou inspecter les lieux. Ils sont employés dans la surveillance et le gardiennage pour la détection d'intrusion. Dans l'armée, les drones militaires sont envoyés à la place des soldats pour effectuer des missions dangereuses.

-Loisirs et éducation : les robots jouets pour enfants ont été vendus dès les années 90, dotés de plusieurs capteurs pour interagir avec leurs environnements, ces robots peuvent prononcer des mots, marcher, faire des mouvements complexes, etc. En éducation, des robots ont été développés pour donner des cours à distance aux élèves en les connectant avec leur professeur.

1.4. Motivation

Nous nous plaçons dans un contexte, où le marché des robots est en pleine explosion, avec une robotique qui conquiert de plus en plus plusieurs aspects de notre vie. Tous les indicateurs nous renseignent sur le futur brillant que va avoir la robotique [6] ; la concurrence mondiale exige la modernisation des installations de production par la robotique, L'efficacité énergétique et les nouveaux matériaux, exigent le rééquipement de la production par des robots sophistiqués, la croissance des marchés de consommation nécessite l'expansion des capacités de production qui peut être satisfaite seulement par une industrie robotisée, l'amélioration de la qualité nécessite des systèmes robotisés de haute technologie, les performances des robots en précision, en efficacité et en réduction des coûts, accroissent la demande en robot, surtout avec les possibilités qu'ils offrent, en effectuant efficacement des tâches compliquées ou dangereuses à la place de l'humain. En outre, d'autres marchés vont se transformer et s'intégrer dans le marché de robotique, par exemple, les industriels de l'automobile travaillent sur des véhicules autonomes et intelligents sans conducteur à l'instar des métros automatiques. Egalement, la population mondiale est vieillissante et a besoin d'une robotique de service pour se maintenir. Enfin, l'intensité des recherches scientifiques dans la

robotique, soutenue par des investissements gouvernementaux et privés, reflète l'intérêt grandissant dans ce domaine.

Le marché de la robotique est mature et il va devenir dans un futur proche saturé comme celui de l'automobile. En conséquence, cette situation nous incite à penser aux enjeux environnementaux auxquels l'humanité fera face, et qui sont :

- 1- Le développement, l'utilisation ainsi que toutes les autres phases du cycle de vie du robot, nécessitent des ressources matérielles et énergétiques extraites de l'environnement. Au même temps, ces ressources s'appauvrissent et elles ont atteint des seuils critiques, pour certaines qui ne pourront pas satisfaire les demandes des générations futures. Donc comment peut-on réduire la consommation de ces flux entrants pour éviter la disparition des ressources naturelles?
- 2- Le parc des robots en fin de vie sera du nombre de plusieurs centaines de million de machine, hors d'usage, voire de l'ordre de milliard. Comment peut-on gérer ces déchets et les revaloriser ?
- 3- Les activités du cycle de vie d'un robot génèrent des flux de sortie vers l'environnement, qui le polluent, comment peut-on réduire ces déchets ?
- 4- Les robots vont interagir avec l'humain, faire partie de sa vie, vont être déployés dans notre environnement et accéder à des lieux délicats, etc. Ces faits exigent que la composition du robot ne doive pas être dangereuse pour l'humain et son environnement.

Pour résoudre ces problèmes, il faudrait intégrer les principes du développement durable et d'écoconception dans les phases du cycle de vie du robot du berceau à la tombe, au niveau matériel et logiciel. Il faudrait aussi, pouvoir estimer les impacts environnementaux par rapport aux choix possibles des parties prenantes, en construisant des scénarios pour adopter le meilleur, qui a moins d'impacts sur l'environnement, qui consomme moins de ressources énergétiques et matérielles, et qui respecte les directives environnementales auquel il est soumis.

Notre travail se déroule dans ces perspectives. Nous proposons un méta modèle de la robotique durable, qui permettra de développer et concevoir des robots durables par la prise en compte des aspects du développement durable. Notre méta modèle permettra de développer des ateliers de génie logiciel, qui serviront à simuler des scénarios de conception de robot et choisir le plus durable.

1.5. Hypothèses de travail

Notre travail est soutenu par les hypothèses suivantes :

- 1- L'univers est un super système complexe, qui comporte un système composé de plusieurs entités qui ont des comportements et qui interagissent. Ces entités (robot, humain, etc.) sont constituées de 2 parties indissociables ; une structure immatérielle représentée par l'information et qui échappe aux lois physiques, et une structure matérielle, qui est tangible et soumise aux lois de la physique. Cette dernière incorpore, exécute et transporte la structure immatérielle, qui à son tour donne un sens existentiel à la structure matérielle.
- 2- L'environnement est menacé ; ses ressources énergétiques et matérielles peuvent s'épuiser, la pollution de l'environnement par l'être humain et ses déchets lui sont nuisibles.
- 3- Le marché de la robotique va devenir prochainement, un marché dominant, avec des productions colossales, envahissant tous les aspects de la vie humaine. Cette situation augmentera la consommation des ressources matérielles et énergétiques et produira des déchets.
- 4- L'environnement existe avant la création du robot, continue d'exister tout au long le cycle de vie du robot et demeurera existant lorsque le robot disparaîtra.
- 5- Le robot est constitué d'une partie logicielle (immatérielle) incorporée dans une partie matérielle. Ces deux parties produisent des impacts sur l'environnement et échangent avec lui des flux énergétiques et matériels.

1.6. Synthèse du chapitre 1

Nous avons introduit dans ce chapitre les idées générales qui sous-tendent notre travail de recherche, l'aspect de la complexité liée à la frontière de la robotique, son évolution qui est le résultat d'intenses innovations scientifiques, et les domaines dans lesquels elle intervient. Ces derniers sont en rapport avec l'expansion de la robotique et ses possibilités.

Les raisons qui nous motivent dans ce travail, notre objectif et les hypothèses sur lesquelles se basent notre recherche, nous poussent à en savoir plus sur le contexte de la robotique durable, spécialement en France. Cela est l'objet du chapitre suivant.

Chapitre 2

Contexte de la robotique durable

Pour connaître le contexte de la robotique durable, nous présentons dans ce chapitre, les travaux trouvés dans ce domaine et les résultats d'une enquête que nous avons menée sur les pratiques environnementales des industriels Français de robotique. Les résultats de ces investigations dans le monde académique et industriel vont nous renseigner sur la situation globale de la robotique durable.

2.1. Etat de l'art de la robotique durable

Après avoir fait des recherches intensives sur les publications sur la robotique durable en français et particulièrement en anglais, nous n'avons pas à ce jour trouvé de travaux de recherche concrets dans ce domaine. Les seuls travaux s'en rapprochant, appartiennent à l'institut Fraunhofer, et sont dans le domaine de la mécatronique, domaine dont la robotique fait partie.

Dans Les recherches de [7], un modèle de systèmes mécatronique qui intègre des aspects du de l'environnement a été développé. Le modèle permet au concepteur de faire une simulation de son système mécatronique selon plusieurs scénarios, pour choisir la solution qui convient au contexte. Le modèle proposé est constitué de plusieurs vues (voir figure 2), élaborées en sous-modèle qui modélisent les aspects importants du système mécatroniques, y compris l'environnement. Ces vues sont décrites comme suit :

- Les exigences (*Requirement*) : cet aspect présente les exigences et les spécifications du système qui doivent être respectées lors du développement (comme la géométrie, les données de qualité, etc.). Chaque exigence (spécification) est décrite verbalement et, si possible, concrétisée par des attributs quantifiables avec leurs caractéristiques.
- L'environnement : ce sous-modèle décrit les conditions techniques du système qui lui permettent de s'intégrer dans son environnement, les influences pertinentes sur le système (telles que le climat environnant, rayonnement thermique, etc.), les Influences perturbatrices qui sont considérées comme des variables de perturbation, les imbrications entre les influences, les événements qui provoquent un changement d'état du système et les catalogues qui permettent de créer des modèles de l'environnement.
- Les fonctions : elles représentent les fonctionnalités que doit remplir le système pour accomplir ses tâches. Une fonction dans le sous modèle de fonction transforme une action en entrée en une action de sortie.
- La structure active : elle définit les éléments du système, leurs attributs (configurations) et les relations entre eux.
- La forme (*Shape*) : cette vue permet de modéliser l'aspect corporel du système en utilisant des outils de CAO en 3D.
- Les scénarios d'application : ils mettent en scène un problème qui doit être résolu dans des cas particuliers, et ils décrivent également à peu près la solution possible, selon les comportements du système. Dans un scénario, les états du système et leurs transitions dans une situation particulière sont initialisés en leurs appliquant les évènements qui peuvent survenir.
- Les comportements: ils sont caractérisés par les états que peut avoir le système, leurs transitions et les actions qui résultent du passage d'un état à un autre.

Ce modèle ne caractérise pas toutes les relations qui peuvent exister entre le système et l'environnement dans toutes les phases du cycle de vie du produit. Il est difficile d'estimer directement les impacts environnementaux et les quantités de flux de matière échangées avec l'environnement, sans faire appel à des outils ou des modèles autres que celui-ci. Les notions d'impacts environnementaux ne sont pas prises en compte car l'environnement est considéré comme un milieu physique qui doit être maîtrisé. En plus, le modèle ne met pas en valeur toutes les phases du cycle de vie du produit, il s'intéresse seulement à la phase d'utilisation, en la simulant selon des scénarios dans la phase de conception, afin de choisir les choix les plus adéquats et en tenant compte des exigences. Néanmoins, certains sous-modèles peuvent servir à optimiser les caractéristiques environnementales du système, surtout au niveau de la consommation énergétique, à titre d'exemple, la structure active admet 2 types de flux entre les éléments du système : des flux énergétiques et des flux d'information sont échangés entre ces éléments. En enrichissant ce sous-modèle, il sera possible de quantifier les flux énergétiques, repérer les zones qui consomment plus d'énergie et les optimiser.

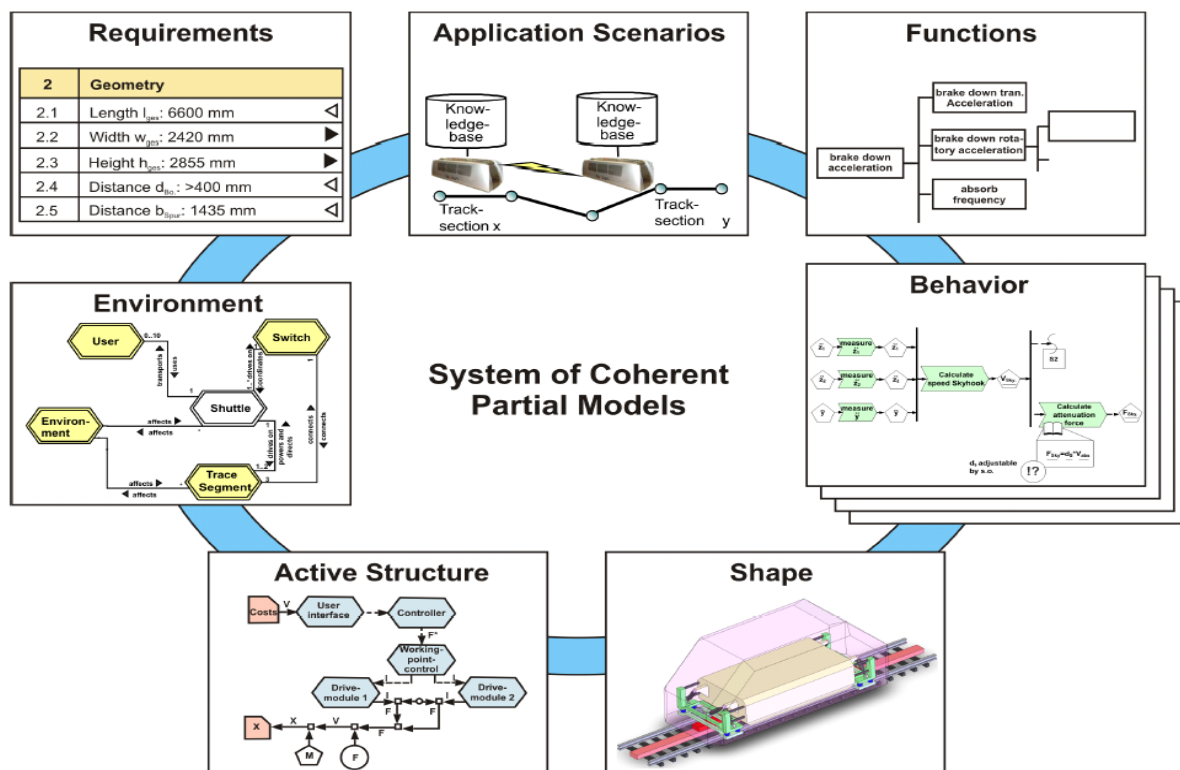


Figure 2 : Le modèle proposé par l'institut Fraunhofer. [7]

2.2. Enquête sur la robotique durable en France

Nous avons effectué une enquête sur les pratiques environnementales des 34 plus grands industriels français en robotique (voir le tableau 1 pour la liste des entreprises évaluées), l'enquête comprend 19 questions, classées en deux catégories: Informations générales et développement durable. Parmi ces questions, nous avons voulu savoir si les principes d'écoconception sont intégrés dans la phase de conception du robot, si les impacts du logiciel sont pris en compte, s'il y a des efforts de réduction de sa consommation énergétique, quelles sont les directives environnementales que les entreprises respectent, etc. (voir dans l'annexe pour la liste complète des questions du sondage). L'enquête a été créée en utilisant l'outil de sondage en ligne *Lime Survey* via le portail de l'UBO. Nous l'avons envoyée aux contacts principaux des 34 entreprises, en mettant en place un intervalle d'un mois (entre Février 2014 et Mars 2014), pour recevoir les réponses. Après la fin de l'intervalle, l'enquête a été fermée et nous avons extrait les données dans un fichier MS Excel.

Abankos	Aldebaran-Robotics	Awa Bot	BAsystemes	CEA-LIST	CRIIF
Delair-Tech	Deltadrone	Eca-robotics	EdF-Intra	Effistore	EOS innovation
EOS-Innovation	First Class Robotic	Induct-Technology	Irstea	Mécanuméric (Charlyrobot)	Medtech
M-TecksEAC	Novadem	Overdriverobotics	R&D Tech	RB3D	Robopec
Robopolis	Robosoft	Robotswim	Staubli	Tecdron	Tecknisolar
Violet	Vitirover	Wanyrobotic	Zodiac-Poolcare		

Tableau 1: les principales entreprises de robotique en France.

Une seule entreprise a répondu partiellement à l'enquête, trois autres ont consultés l'enquête, sans lui prêter attention, et le reste des entreprises n'ont même pas pris la peine de consulter le sondage. Le taux de participation était d'environ 3% et l'intérêt et de pertinence de l'enquête était d'environ 12%.

La seule entreprise qui a répondu à l'enquête est la R & D Tech France. La personne ayant répondu, nous a informé qu'ils interviennent dans le domaine militaire, qu'ils ont une stratégie intégrée de l'environnement, qu'ils sont certifiés ROHS, qu'ils travaillent avec quelques principes d'écoconception afin de réduire la consommation de matières premières et qu'ils utilisent des matériaux recyclables dans leurs robots. Cependant, ils n'évaluent pas les indicateurs environnementaux ni effectuent l'analyse du cycle de vie de ses robots. Ils indiquent qu'ils prennent en considération la modularité, maintenabilité et la fiabilité, au niveau logiciel et matériel et qu'ils ne publient pas les données de fiabilité.

Pour tenter d'interpréter ces maigres résultats, nous avançons les hypothèses suivantes:

1. Peut-être que certains des messages ne sont pas arrivés à destination parce que nous nous sommes trompés de destinataires, ou bien parce qu'ils étaient considérés comme des spams par le filtre anti-spam dans les boîtes de messagerie des entreprises.
2. En raison de la sensibilité de leurs réponses (dans le cas où ils n'intègrent aucun aspect du développement durable) et pour préserver la réputation de leurs entreprises, ils ont décidé d'ignorer l'enquête. Dans ce cas, ça signifie que 97 % des entreprises négligent les préoccupations environnementales et ne les mettent pas en œuvre dans leurs stratégies.
3. Certaines données sont confidentielles, même si les entreprises considèrent les impacts environnementaux, elles préfèrent ne pas divulguer ces informations qui peuvent servir contre eux par leurs concurrents par exemple.

2.3. Synthèse du chapitre 2

Le contexte de la robotique durable est particulier ; il semblerait d'après nos investigations et l'enquête que nous avons accomplie, que le sujet est prématuré dans le monde académique et au niveau industriel. Malgré le besoin intrinsèque de passer d'une robotique classique, dont la conception et le cycle de vie ne considèrent pas les aspects environnementaux à une robotique éco-intelligente (ou durable), permettant de soutenir les besoins des futures générations. De toute façon, ce besoin se globalisera et se renforcera dans un futur proche, puisque il y a plusieurs signes annonciateurs et les pièces du puzzle dans le domaine de la robotique, du génie logiciel durable, de l'économie et du développement durable auront seulement besoin d'être assemblées et intégrées les unes aux autres.

Chapitre 3

Analyse du domaine de la robotique durable

La robotique, dans laquelle interviennent la mécanique, l'électronique et l'informatique, permet de développer des robots composés d'une partie logicielle incorporée dans une structure matérielle. Ces derniers forment des ensembles homogènes et cohérents, pourvus d'aptitudes de perception, de contrôle, de communication et de décision. Toutefois, L'intégration de ces deux couches (matérielle et logicielle) pose aux chercheurs de nouveaux défis complexes, comme la question du rôle de la couche logicielle dans l'utilisation des ressources matérielles et la consommation d'énergie électrique, ou bien l'impact des choix de conception et des autres phases de cycle de vie du robot sur l'environnement et aussi la prise en compte de contraintes comme les limitations matériels dans le développement logicielle, le devenir du robot lors de la fin de sa vie, etc.

Dans ce chapitre nous effectuons une analyse du domaine de la robotique, tout en insistant sur les liens avec les aspects de développement durable et de prise en compte des enjeux l'environnement pour mettre en avant la robotique durable.

3.1. Architecture des robots

Un robot est une entité composée principalement d'une structure mécanique (locomotion, manipulation et un corps qui supporte et réunit différentes parties), d'un système électronique pour la perception, l'exécution des instructions de contrôle et l'action, et d'une couche logicielle pour le contrôle, le traitement des données et la prise de décision.

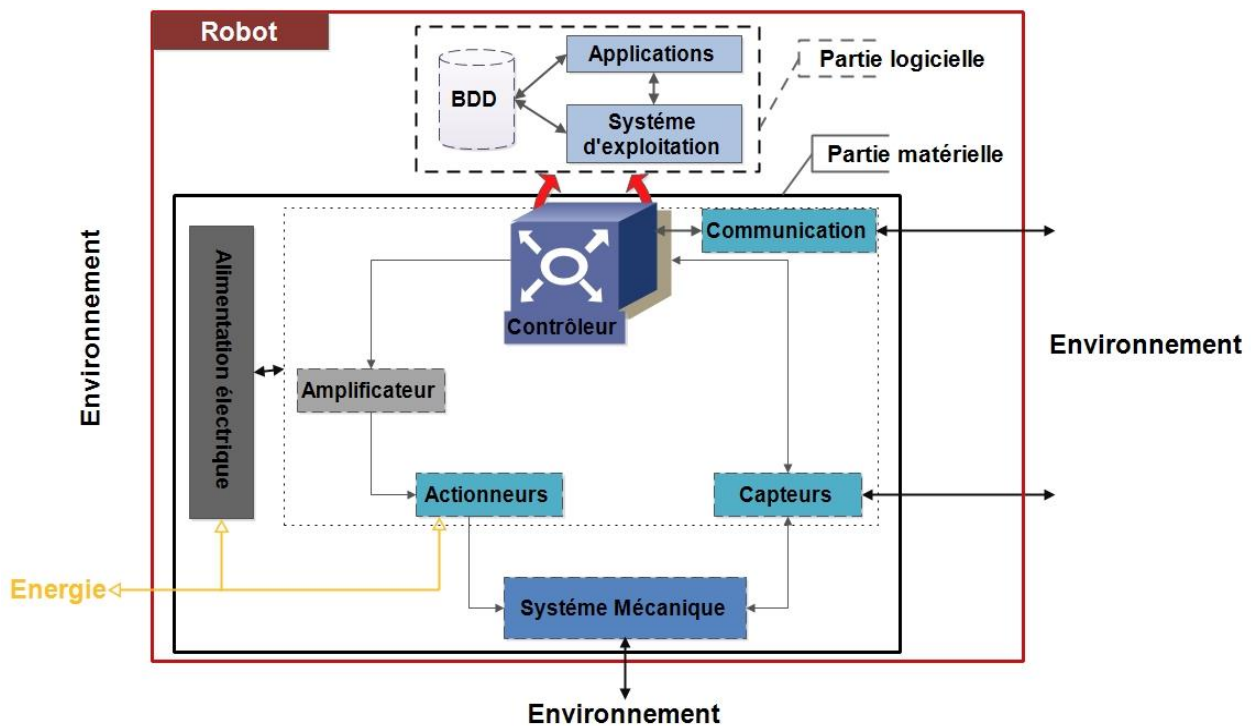


Figure 3 : Architecture Générale d'un robot.

La figure 3 décrit l'architecture d'un robot. Les capteurs collectent des informations sur l'environnement et l'état du système. Ces informations peuvent être des données de proximité, de vitesse, de température, de force, de vibration, etc. Elles sont produites sous forme de signaux numériques ou analogiques par les capteurs. Ces derniers peuvent être conditionnés. Après, ils sont transférés sous forme de signaux numériques (les signaux analogiques sont convertis en signaux

numériques) vers le contrôleur. Ce dernier est considéré comme le cerveau du robot, il traite les signaux reçus des autres composants et exécute la couche logicielle du robot. Le contrôleur peut être un PC (micro-ordinateur) composé d'un microprocesseur, d'une mémoire, de périphériques d'entrée et de sortie (écran, lecteurs, haut-parleurs, etc.), et de dispositifs (disque dur, cartes d'extension, routeur, etc.), ou bien, il peut être aussi, une unité microcontrôleur (MCU) constituée d'un processeur, d'une mémoire et d'interfaces d'entrées-sorties (Led, relais, solénoïdes, etc.) sur une même puce. Le contrôleur produit des signaux d'instruction qui sont convertis en signaux analogiques et sont amplifiés dans certains cas avant d'atteindre les actionneurs. L'actionneur peut être considéré comme un muscle artificiel, il convertit les signaux électriques en mouvement, ou en action mécanique utile [8], entraînant le mouvement de corps mécaniques. Enfin le système mécanique est composé d'assemblages de plusieurs corps connectés par des liaisons. Il peut prendre la forme de manipulateurs ou de plateforme de locomotion.

3.2. La partie matérielle

Elle est composée d'une structure mécanique, d'une structure électronique et d'une alimentation électrique. La partie matérielle est fabriquée à partir de matériaux qui proviennent de l'environnement ; les ressources naturelles sont extraites de l'environnement, transportées vers des usines pour transformation, assemblage et fabrication en produits, avant d'être distribuées pour utilisation.

3.2.1. Le châssis

Le châssis doit être résistant aux dangers physiques extérieurs comme les chocs les frottements ou les vibrations, pour protéger les composants matériels du robot et les maintenir hors exposition. Il doit être aussi rigide et capable de supporter tous les autres composants matériels. Sa géométrie permet de définir l'esthétique et la forme de base du robot. Celle-ci est définie selon les fonctions que doit remplir le robot et les contraintes auxquelles le robot sera soumis. Le poids du châssis influence la consommation énergétique du robot ; plus les matériaux le composant sont solides et consistants, plus le robot est lourd. Ce poids additionnel, sollicite plus d'énergie nécessaire pour bouger le robot et donc augmente la consommation énergétique du robot. Les innovations récentes en science des matériaux, tels que les alliages de titane et les composites, ont permis de trouver de nouveaux compromis, néanmoins ces matériaux sont encore très coûteux [9]. Pour construire des robots durables, les matériaux composant le châssis doivent respecter certaines caractéristiques environnementales, par exemple, ils doivent être non toxiques, réutilisables, recyclables, issus de ressource renouvelable ou bien d'autre matériaux recyclés.

3.2.2. La structure mécanique

Elle permet au robot de se déplacer ou de manipuler des objets. Elle est composée de plusieurs corps mécaniques connectés par des liaisons.

- a) Corps mécaniques (*Links*): Un corps est une partie matérielle rigide dotée d'au moins d'un nœud de liaison (un trou qui soutient la fixation des corps). Ils peuvent être classifiés (voir figure 4) selon le nombre de nœuds qu'ils contiennent en corps unaires, binaires, ternaires, quaternaires et N-aires où le N correspond au nombre de nœuds [10]. Un exemple typique est le tronc d'un robot humanoïdes qui contient 5 nœuds ; 2 nœuds connectés chacun à un bras, 2 autres connectés chacun à une jambe et le dernier relié au cou.

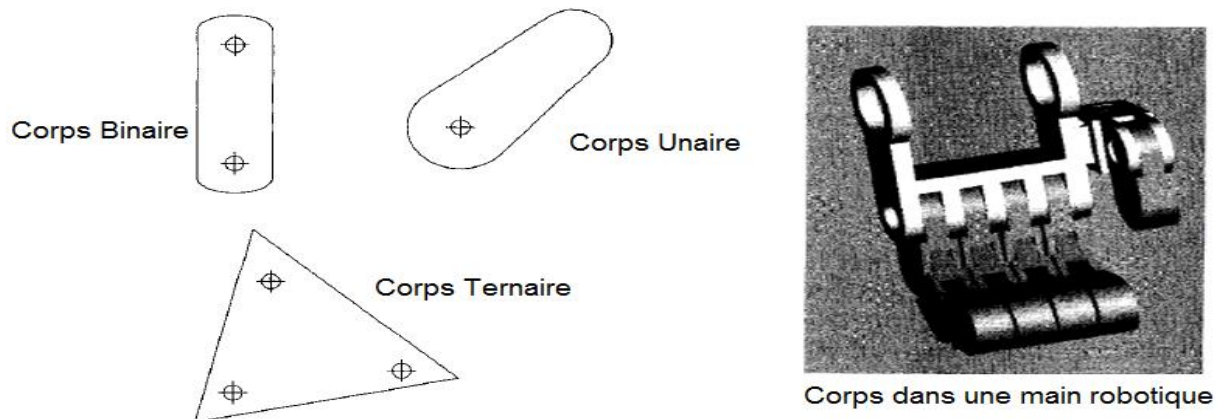


Figure 4 : Corps mécaniques d'un robot. [10]

- b) Liaisons mécaniques (*Joints*) : une liaison est une articulation qui connecte deux ou plusieurs corps à leurs nœuds. Elle limite les mouvements des corps connectés (voir figure 5). Si une liaison relie seulement deux corps, elle est aussi appelée une paire cinématique. Les liaisons peuvent être classées selon les degrés de liberté (*Degree Of Freedom*) permis par la paire cinématique [10].

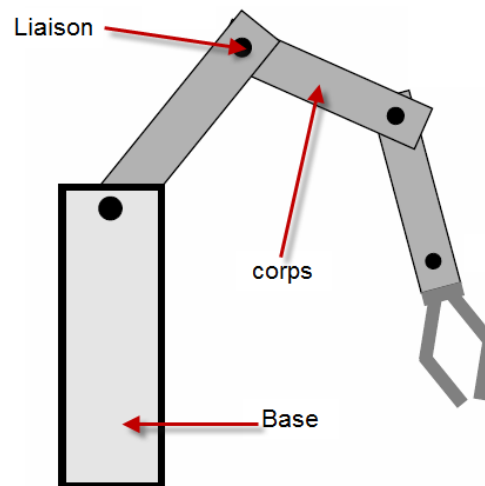


Figure 5: Articulations d'un bras robotisé. [11]

Plusieurs types de plateformes mécaniques existent, comme :

1. Les Manipulateurs : Ce sont des mécanismes qui permettent de manipuler des objets. Ils sont utilisés pour le serrage, le levage et le déplacement d'objets avec plusieurs degrés de liberté. Ils sont composés de 2 sous-ensembles mécaniques :
 - Les bras robotisés : ils sont composés de plusieurs corps mécaniques connectés en série par des articulations.
 - Les effecteurs d'extrémité : ils se trouvant à l'extrémité du bras, ces dispositifs imitent les doigts et permettent d'interagir avec l'environnement.
2. Les plateformes locomotrice : Ce sont des mécanismes qui permettent au robot de se déplacer dans l'espace d'un point à l'autre, ils assurent la mobilité du robot, parmi les types de plateformes locomotrices, il y a :
 - Les plateformes à roues.
 - Les plateformes à chenilles.
 - Les plateformes à pattes.

3.2.3. Les actionneurs

Un actionneur est un dispositif qui déplace un ou plusieurs corps mécaniques (voir la figure 6) et produit les mouvements des effecteurs d'extrémité [11]. L'actionneur est principalement un convertisseur d'énergie ; il transforme l'énergie qu'il reçoit en action mécanique. Il existe beaucoup de type d'actionneurs ex : actionneurs hydrauliques, à combustion, pneumatiques, moteurs électriques, etc.

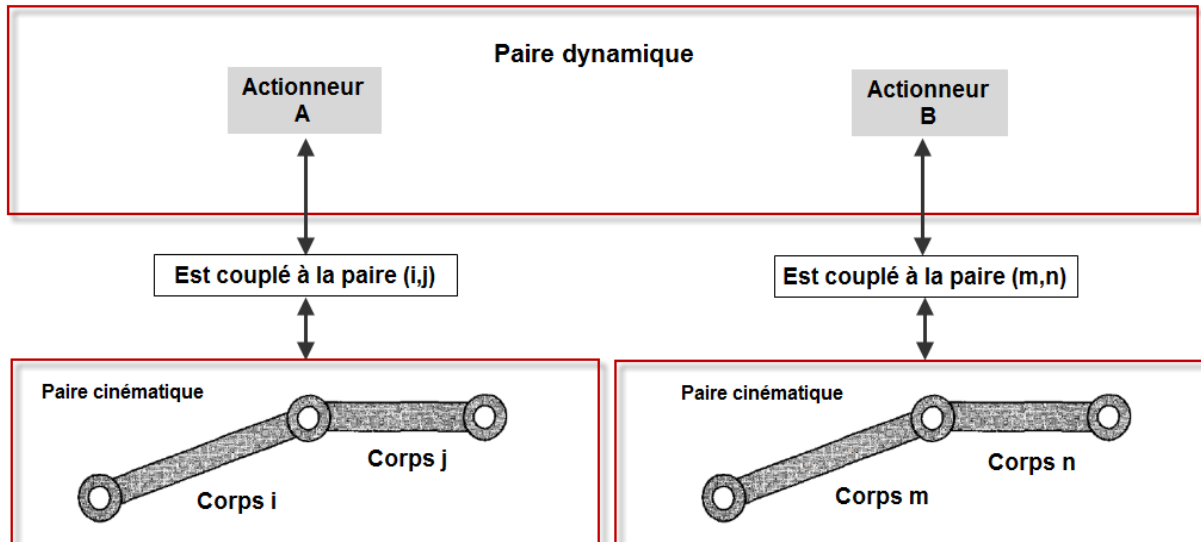


Figure 6 : Relation entre les actionneurs et les corps mécaniques. [10]

La qualité des actionneurs est caractérisée par les paramètres suivants [12] :

- Puissance continue en sortie (*Continuous power output*) : la force ou le couple maximum accessible en permanence sans dépasser les limites de température.
- Amplitude du mouvement (*Range of motion*): l'intervalle linéaire ou rotatif de mouvement.
- Résolution (*Resolution*): l'incrément minimal de la force ou du couple réalisable.
- Précision (*Accuracy*): la linéarité de la relation entre l'entrée et la sortie.
- La force maximale (*Peak force*): capacité maximale d'action de l'actionneur.
- La dissipation thermique (*Heat dissipation*): puissance maximale de dissipation de la chaleur en régime permanent.
- Caractéristiques de vitesse (*Speed characteristics*): force (ou couple) en rapport avec la vitesse.
- Vitesse à vide (*No load speed*): vitesse sans charge externe
- Fréquence de réponse en (*Frequency response*): la gamme de fréquence sur laquelle la sortie suit l'entrée.
- Alimentation (*Power requirement*): type de puissance (AC ou DC), le nombre de phases, niveau de tension, et la capacité du courant.

3.2.4. Les capteurs

Un capteur est un dispositif de mesure et un convertisseur de quantité physique. Il transforme en signal qui peut être lu par le contrôleur, les modifications physiques survenus dans l'environnement, afin de permettre au robot d'avoir une perception du monde qui l'entoure. Il peut être classé selon plusieurs critères :

- La position : elle concerne l'emplacement du capteur. Ce dernier est local s'il est placé dans le corps du robot ou bien, global s'il est installé hors du robot, dans l'environnement.

- Le cadre: il définit le champ d'application du capteur. Le capteur proprioceptif collecte les données des composants du robot, et le capteur extéroceptif collecte les données du milieu extérieur au robot.
- L'interaction: elle concerne l'influence du capteur sur son milieu; le capteur actif agit sur l'environnement du robot. Le capteur passif collecte les données sans affecter le milieu.
- Le fonctionnement: selon son type de fonctionnement, un capteur peut être classé en [10]:
 - Capteur électrique: capteur qui transforme une grandeur physique ou chimique directement en un signal électrique correspondant.
 - Capteur électromagnétique: capteur qui utilise le champ magnétique pour faire la conversion du signal.
 - Capteur optique: capteur utilise la lumière en tant que médium pour effectuer la conversion du signal.

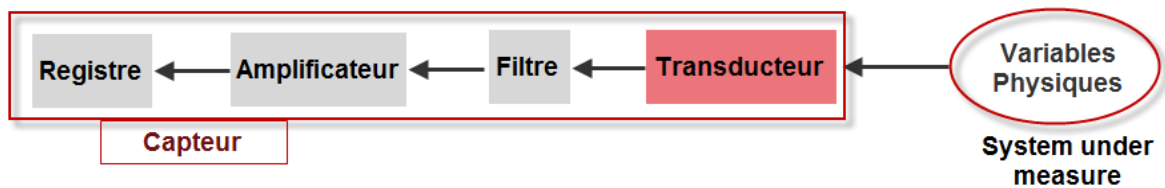


Figure 7 : Principe général de fonctionnement d'un capteur.

Le principe de fonctionnement des capteurs en robotique est décrit dans la figure 7. Les variables physiques mesurées par le capteur (ex. position, vitesse, force, etc.) sont converties en différents signaux électriques (ex. tension, résistance, etc.) par le transducteur, idéalement sans perte d'énergie. Par exemple, le capteur de vitesse d'un moteur ne devrait pas consommer d'énergie mécanique du moteur lui-même [10]. Ensuite le signal converti est conditionné par un filtre et un amplificateur avant d'être enregistré dans un registre.

La qualité d'un capteur est caractérisée par plusieurs attributs [8]:

- a. *Caractéristiques statiques*: ces paramètres sont indépendants du temps et caractérisent le signal en sortie du capteur. Les différentes caractéristiques appartenant à cette catégorie sont :
 - *L'intervalle (Range)* : définit la plage admissible de la quantité physique pouvant être détectée par le capteur, se trouvant entre un minimum et un maximum.
 - La Précision (*Accuracy*): La différence entre la valeur mesurée véritable et réelle.
 - La Sensibilité (*Sensitivity*) : La relation entre l'entrée mesurée et la sortie du capteur.
 - La Résolution (*Resolution*) : La plus petite variation de la valeur d'entrée qui peut produire un changement observable de valeur dans la sortie du capteur.
 - L'Hystérésis (*Hysteresis*) : La différence maximale de sortie du capteur pour la même quantité d'entrée, avec une mesure effectuée en augmentant l'entrée à partir de zéro et une autre en diminuant de l'entrée à partir de la mesure maximale.
 - La Répétabilité (*Repeatability*): L'erreur dans la valeur de sortie pour l'application répétée de la même valeur d'entrée. Plus l'erreur diminue, plus la précision du capteur augmente (la précision est l'inverse).
 - La Stabilité (*stability*) : Se réfère à la variation de la sortie lorsque la quantité d'entrée ne change pas.
- b. *Caractéristiques dynamiques* : ces paramètres sont dépendants du temps. Ils décrivent les caractéristiques du capteur depuis l'instant où la variable physique a changé jusqu'à l'instant précédent la génération du signal de sortie. Parmi ces paramètres :
 - *Le Temps de montée (Rise Time)* : le temps nécessaire pour changer la valeur de sortie d'un certain pourcentage à un autre.

- Constante de temps (*Time Constant*) : est le temps nécessaire pour atteindre la sortie de 63,2% de la valeur finale. Une grande constante de temps implique un capteur lent, tandis qu'une avec une petite valeur indique un capteur qui répond rapidement.
- Durée de stabilisation (*Settling Time*) : le temps qu'il faut pour que la valeur en sortie atteigne le pourcentage du régime permanent.
- Bande passante (*Bandwidth*): définit la plage de fréquence dans laquelle le capteur est conçu pour fonctionner.

3.2.5. Le contrôleur

Le contrôleur est le composant électronique chargé de commander tous les opérations qu'effectue le robot. Il permet d'exécuter les instructions de la couche logicielle, d'assurer l'interaction de cette couche avec la partie matérielle et de gérer le fonctionnement de tous les autres composants matériels qui lui sont rattachés. Le contrôle de ces différentes fonctionnalités s'effectue par l'exécution de programmes informatiques. De plus, chaque capteur, actionneur ou dispositif matériel programmable, a son propre code informatique incorporé dans le contrôleur qui permet de le gérer. Un contrôleur est composé des éléments suivants [8]:

- Le processeur (CPU) : est un dispositif électronique qui remplit le rôle d'unité centrale de traitement, il exécute les programmes informatiques sous forme de codes de bas niveau en langage machine. Il comprend les 3 éléments de base suivants:
 - L'unité de contrôle (CU).
 - L'unité arithmétique et logique (ALU).
 - Les registres.
- La mémoire: est un dispositif à semi-conducteurs, utilisé par le processeur pour stocker les données.
- Les Bus : sont des dispositifs électroniques qui permettent de transférer les données entre le processeur et les autres dispositifs électroniques
- En robotique, on peut employer 2 types de contrôleurs :
- *Un micro-processeur* : composé d'un CPU installé sur une carte mère, où se trouvent aussi les supports de mémoire, les cartes d'extensions et les interfaces d'entrée/sortie ... etc., le microprocesseur communique avec tous ces composants par des bus externes qui passent par la carte mère.
- *Un Microcontrôleur* : circuit intégré qui concentre le processeur, la mémoire et le système d'entrée/sortie en un circuit unique.

Les microprocesseurs consomment beaucoup plus d'énergie électrique que les microcontrôleurs parce qu'ils demandent beaucoup plus de capacité mémoire et ont une fréquence d'exécution plus élevée. Selon [8] , dans les microprocesseurs, la taille de la RAM est typiquement dans les *Giga-octets* et la fréquence d'horloge est dans la gamme de *GHz*, tandis que dans les microcontrôleurs la taille mémoire est entre 1 à 100 *K octets* et la fréquence d'horloge est en dessous de *GHz*.

3.2.6. Vers une structure matérielle durable

L'intégration des principes d'écoconception dans le développement de la partie matérielle du robot permet de concevoir des robots durables. Cette intégration se fait en sélectionnant les matériaux des composants matérielles qui ont moins d'impacts environnementaux et en intégrant l'évaluation des échanges des phases du cycle de vie du robot lors de la conception pour estimer leurs impacts et choisir les meilleures alternatives.

En premier lieu, le concepteur a besoin de base de données qui le renseigne sur les caractéristiques et les impacts environnementaux des matériaux qui veut utiliser, sur les directives que ces matériaux respectent (certains sont interdits par les lois) et sur leurs valorisations en fin de vie.

Ensuite, il intègre principalement l'analyse de cycle de vie du robot dans la conception pour pouvoir estimer les échanges de flux de matière et d'énergie avec l'environnement et les impacts environnementaux selon des scénarios liés aux choix qu'il a fait .

Finalement, il devra en choisir l'alternative qui permet d'économiser de l'énergie (notamment dans la phase d'utilisation), la consommation de matières, de réduire les déchets des phases de cycle de vie du robot et de valoriser le robot en fin de vie par le recyclage, la biodégradation et la réutilisation.

L'analyse de cycle de vie d'un produit (ACV) est une approche d'évaluation environnementale qui permet d'estimer les impacts d'un produit (dans notre cas le robot) sur l'environnement à chaque phase de son cycle de vie, depuis l'extraction des matières premières qui le composent jusqu'à sa mise au rebut (voire la figure 8). C'est aussi, la méthode la plus aboutie en termes d'évaluation globale et multicritère [13]. L'ACV a été normalisé par l'ISO14040, elle se réalise en quatre étapes [14] [15] [16] [17]:

- 1- La définition des objectifs et du champ d'étude : Cette étape permet d'explicitier le cadre de l'analyse, ses objectifs et sa portée. Parmi les concepts que cette étape doit fournir, on mentionne :
 - L'unité fonctionnelle: elle détermine les fonctions du système, elle est choisie selon les services que le produit fournit. Elle est accompagnée de flux de référence qui désigne la quantité du produit pris en compte et de consommables utilisés.
 - La frontière du système: elle est définie par les étapes principales du cycle de vie du produit, ses processus élémentaires et les flux pris en compte.
 - Les processus élémentaires: le système de produit est décomposé en plusieurs procédés élémentaires (transformation, stockage, transport ... etc.), chacun de ces processus reçoit des flux entrants et fournit des flux sortants vers l'environnement ou des flux intermédiaires vers d'autres processus.
- 2- L'inventaire des flux entrants et sortants du système (LCI) : cette étape consiste à répertorier l'ensemble des flux pris en compte pour l'étude. Ces flux sont associés aux processus élémentaires du système. Deux catégories de flux sont identifiées:
 - Les flux économiques: ils représentent les flux de matière, d'énergie, de services, échangés entre les processus élémentaires et avec des systèmes extérieurs.
 - Les flux élémentaires: ils définissent les flux échangés avec l'environnement par exemple : la consommation matières premières, les rejets dans l'environnement, etc.

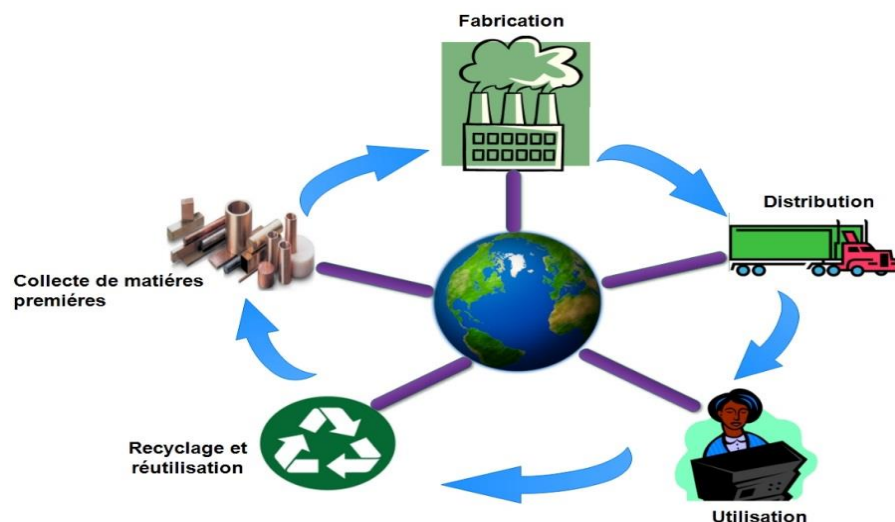


Figure 8: Cycle de vie du matériel du robot.

Les flux associés à chaque processus élémentaire, sont quantifiés pour identifier tous les éléments qui ont un impact environnemental.

3- L'évaluation des impacts (LCIA): elle consiste à traduire les flux de l'inventaire en différentes catégories d'impacts environnementaux. Cette traduction fait appel à des bases de données ACV qui associent à chaque flux des impacts (voir figure 9) qui sont mesurés par des indicateurs environnementaux. Il existe 2 types d'impacts :

- *Impacts mid-point* : Ce sont des impacts orientés problèmes, situés au milieu de la chaîne de cause/effet. Leurs indicateurs mesurent les effets globaux des flux émis ou consommés. Les masses émises ou extraites sont multipliées par des facteurs de caractérisation (définis l'importance d'une substance dans une catégorie) et sont additionnées dans chaque catégorie d'impact mid-point, souvent exprimé en kilogrammes d'une substance de référence. on les calcule selon la formule suivante : $SI_i = \sum_s FI_{s,i} * M_s$
SI_i :Indicateur intermédiaire pour la catégorie d'impact
FI_{s,i} :Facteur de caractérisation intermédiaire de la substance s dans la catégorie i
M_s :Masse émise ou extraite de la substance s.

- *Impacts end-point* : Ce sont des impacts orientés dommages, ils se placent à la fin de la chaîne de cause/effet. Leurs indicateurs estiment les dommages potentiels des flux, en affectant plusieurs catégories d'impacts intermédiaires à une ou plusieurs catégories de dommages. Selon [18] , ces indicateurs sont calculés selon la formule suivante :

$$SD_d = \sum_i FI_{i,d} * SI_i$$

SD_d :Indicateur de dommage pour la catégorie d'impact d.

FI_{i,d} :Facteur de caractérisation de dommage reliant la catégorie intermédiaire i à la catégorie de dommage d.

SI_i :Indicateur de la catégorie intermédiaire i.

4- L'interprétation lors de laquelle lequel l'inventaire des flux entrants/sortants et les impacts environnementaux sont vérifiés, résumés, quantifiés et évalués pour s'assurer que ces 2 phases respectent les objectifs et le champ d'étude définis dans la première étape. Cette dernière phase fournit des recommandations, des restrictions et des conclusions.

Remarque : les impacts mid-points sont difficilement interprétables pour les non spécialistes du développement durable, mais ils maintiennent l'avantage d'être définis à des instants et des lieux précis. Au contraire des impacts end-points qui sont difficiles à situer dans l'espace-temps, en plus, ils peuvent donner lieu à des controverses.

Pour faciliter l'analyse de vie et la collaboration entre différents intervenants dans le cycle de vie du matérielle, des bases de données doivent être créées, pour nous renseigner sur les caractéristiques environnementales des produits et leurs attributs qui influencent l'environnement, parmi les données pertinentes qu'on peut regrouper, il y a :

- La durée de vie des composants matériels.
- Les données de fiabilité : temps moyen entre les pannes (MTBF) et le temps moyen de réparation (MTTR).
- Les qualités environnementales : toxicité, recyclabilité, réutilisabilité, etc.
- Les indicateurs environnementaux pour pouvoir estimer les impacts sur l'environnement.
- Les directives environnementales auxquelles ils sont soumis (ROHS, DEEE, etc.).

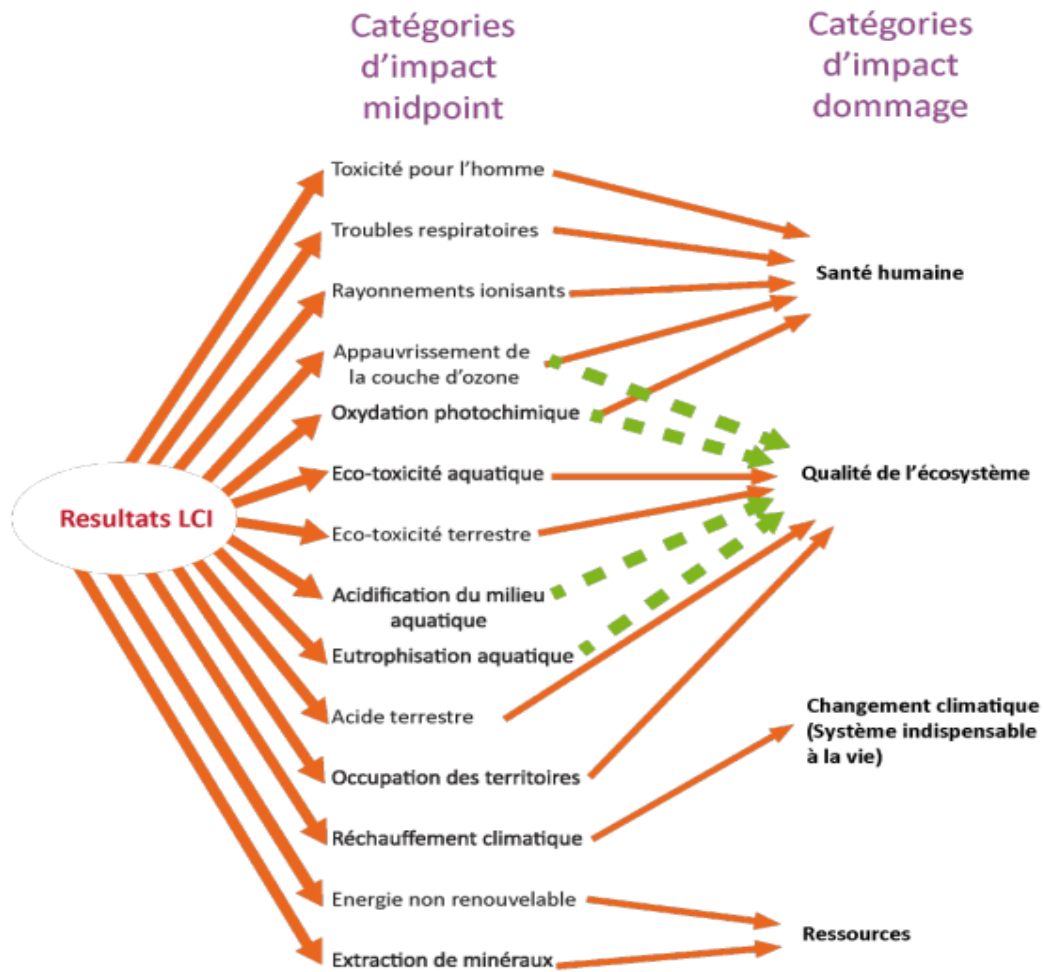


Figure 9: Catégories d'impacts environnementaux. [16]

3.3. La partie logicielle

La partie logicielle du robot est composée des données et des programmes qui permettent au robot de contrôler le fonctionnement de ses composants matériels, d'interagir avec son environnement et d'exécuter ses tâches. Ces dernières font partie des 4 fonctions principales qu'effectue le robot et qui sont : la perception, la communication, la décision et l'action.

Dans la perception, le robot a des données internes sur ses composants matériels pour les activer et reçoit des capteurs les données sur son environnement externe, ou bien sur ses composants matériels associés à ses capteurs. Ces données sont traitées par les programmes informatiques du système et génèrent des résultats qui correspondent à des décisions. Ces dernières sont converties en signaux de commandes et sont communiqués aux composants matériels contrôlables pour qu'ils les exécutent. Les dispositifs matériels traduisent ces signaux de commandes en actions physiques. L'ensemble de ces actions permettent au robot d'adopter un comportement à la situation rencontrée, lors du déroulement de sa mission.

La communication s'effectue physiquement par l'échange de signaux (électriques, optiques, électromagnétiques, etc.), qui encodent l'information (données ou bien les instructions de commandes) et la transmettent au travers de mediums matériels (bus, câbles, etc.) à sa destination. Elle peut être logique, par échange de messages entre les différents composants logiciels, dans ce cas, ces messages sont transformés en signaux avant d'être transmis à leurs destinations.

3.3.1. Les architectures logicielles en robotique

Une architecture de contrôle permet de définir les différents composants logiciels d'un robot, de spécifier leur organisation, leurs interactions et comment ils communiquent. Les principales caractéristiques d'une architecture logicielle en robotique sont [19] :

La Programmabilité: une architecture doit permettre aux robots d'être des machines hautement et facilement programmables tant du point de vue du programmeur de la machine que celui de l'utilisateur final.

L'intégration: l'architecture supporte différents types de composants logiciels et permet d'offrir des mécanismes de communication et d'échange de données entre eux.

La cohérence: En fonction de sa perception de son environnement, le robot fournit des comportements et exécute des actions non contradictoires.

La réactivité: les différents composants de l'architecture doivent être capables de réagir de façon appropriée aux stimuli spécifiques qu'ils reçoivent.

La robustesse: L'architecture doit permettre d'exploiter la redondance des sources d'information, des traitements, et la multiplicité des ressources matérielles.

La sûreté: l'architecture utilise de méthodes qui garantissent la sûreté du fonctionnement du robot.

L'extensibilité: La modularité de l'architecture doit permettre d'ajouter de nouvelles fonctionnalités, sans nuire aux composants existants.

Il existe 3 familles d'architectures logicielles :

a) **Les architectures Hiérarchisées**

Elles se basent sur les principes d'intelligence artificielle. Elles sont organisées en 3 niveaux (ou couches) hiérarchisés (voir la figure 10). Ces niveaux sont [20]:

- *Le niveau fonctionnel* : il contient les modules de perception qui sont chargés de transformer les informations qui proviennent des capteurs en données symboliques. Ce niveau comporte aussi d'autres modules qui permettent de transformer les données symboliques qui proviennent de la couche supérieure en commandes destinées aux actionneurs.
- *Le niveau exécutif* : il permet de superviser les tâches du robot par des modules de supervision qui permettent de gérer la couche fonctionnelle.
- *La couche décisionnelle* : elle permet de définir les décisions et d'établir le plan des tâches à exécuter pour permettre au robot d'accomplir sa mission.

Les données collectées par les capteurs sont diffusées d'une couche à l'autre verticalement, de bas en haut, avec des transformations au niveau de chaque couche, jusqu'à ce qu'elles arrivent à la couche décisionnelle, où elles sont utilisées pour générer les décisions. Ces dernières sont transmises à leur tour de haut en bas, jusqu'à leur arrivée à la couche fonctionnelle, qui les transforme en commandes destinées aux actionneurs. Cette dispersion de l'information reflète la dépendance qui existe entre les couches ; chaque couche fonctionne que lorsqu'elle reçoit les données de la couche inférieure, ou celles de la couche supérieure. Cette dépendance entre couches augmente le délai des actions puisque l'information doit passer par les couches inférieures avant d'arriver à la couche supérieure, ajoutant le temps de production de la décision et son passage vers les deux couches inférieures avant d'arrivée aux actionneurs.

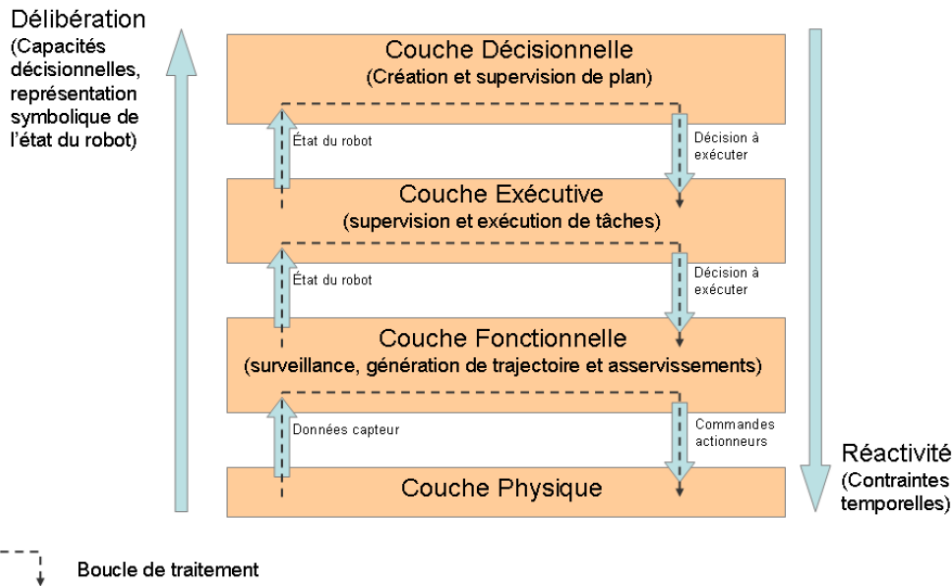


Figure 10 : Le modèle d'architecture hiérarchisée. [22]

b) Les architectures réactives (comportementales)

Elles sont composées de plusieurs modules réactifs ; chaque module possède un comportement réactif qui associe à chaque stimulus reçu d'un capteur une commande destinée à un actionneur. Certains comportements peuvent produire des actions opposées, en générant des commandes contradictoires. Prenant l'exemple d'un robot qui détecte du sucre à côté d'un mur ; la sortie du comportement *détecter_sucré()* est la commande qui permet d'aller vers cette cible pour la collecter (commandes lancées aux actionneurs). Néanmoins, au même temps, le mur constitue un obstacle et le comportement *détecter_obstacle()* a une sortie représentée par la commande d'éviter la cible. Ces 2 commandes se contredisent. Pour résoudre cette incohérence comportementale, un module d'arbitrage a été intégré pour décider quel comportement et quelle commande il faut choisir dans cette situation particulière (voir la figure 11).

La politique d'arbitrage a fait l'objet de plusieurs études. Brooks [21] dans l'architecture de subsumption, a adopté une solution de classement des comportements par des niveaux de compétences, organisés de la plus prioritaire à la moins importante. Chaque module peut supprimer l'entrée ou inhiber la sortie d'un autre module.

L'inconvénient majeur de ce type d'architecture est la difficulté d'adaptation, en effet, puisque tous les comportements du robot sont prédéfinis dans ses modules, celui-ci ne peut pas s'adapter aux nouveaux changements dans son environnement lorsqu'ils lui proposent de nouvelles situations auxquelles les comportements n'existent pas.

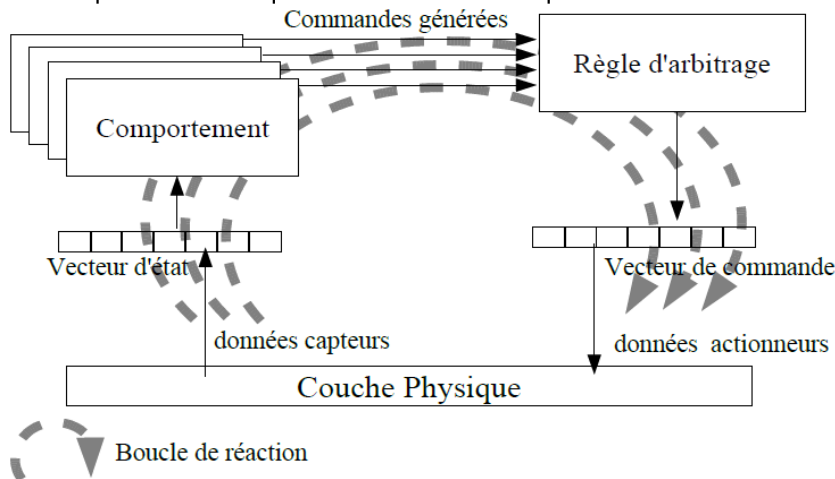


Figure 11 : Le modèle d'architecture réactive. [20]

c) *Les architectures hybrides*

Les architectures hybrides intègrent les concepts de réactivité et de décisions des deux architectures précédentes et permettent de répondre aux problèmes de latence de décision et de manque d'adaptation. Ces architectures intègrent une hiérarchisation des couches permettant la symbolisation des informations et la prise de décision, à cette hiérarchisation s'ajoutent des boucles réactives imbriquées permettant à chaque couche de fournir des réactions adaptées à sa dynamique [22]. Elles se composent généralement de deux parties ; une partie supérieure délibérative, chargée de la planification, la localisation et la navigation et une partie inférieure réactive, qui peut être réalisée sous forme de comportements, formant des boucles sensorimotrices en reliant les actions aux perceptions avec une phase de décision très courte [23].

3.3.2. Les composants logiciels

Les composants logiciels sont utilisés intensément dans la communauté de développement du logiciel et plus particulièrement en robotique, pour développer les applications logicielles du robot. L'approche de développement logiciel à base de composants permet de réutiliser des composants logiciels qui se trouvent dans d'autres applications en les adaptant à un nouveau contexte. Cela produit des gains de temps et de coûts, et améliore le cycle de vie du logiciel et sa durabilité environnementale ; les consommations et les déchets générés lors d'un cycle de vie logiciel sans réutilisation de composant sont évités.

Les composants sont mis en place généralement par des middlewares, chaque composant est principalement constitué des mécanismes suivants [20] :

- Un mécanisme de composition : il permet de mettre en relation au moins deux composants, afin qu'ils réalisent une ou plusieurs fonctionnalités d'une application.
- Un mécanisme d'exécution : il interprète le code des composants, afin que ces derniers réalisent leurs fonctionnalités.
- Un mécanisme de déploiement : il permet de rendre des composants prêts à être exécutés.
- Un mécanisme de communication : il permet à au moins deux composants d'échanger des messages.

Afin d'étendre notre compréhension des composants, nous décrivons dans ce qui suit leurs notions principales qui les définissent, en se basant sur la taxonomie (voire la figure 12) et les explications de [24] :

- L'interface service (*Service Interface*) : elle définit les services que doit fournir un composant. Ces derniers découlent d'une abstraction des comportements attendus. Ils sont spécifiés en termes d'opérations, les paramètres de ses opérations, leurs types et leurs sémantiques.
- Les métas information (*Meta Information*) : elles fournissent les informations qui décrivent les composants, par exemple : la liste des ressources qu'utilise un composant, le nom de ses interfaces, etc.
- Les ressources (*Ressources*) : elles spécifient tous les moyens que le composant nécessite pour fonctionner, par exemple : connexions aux bases de données, liste des interfaces service des autres composants, etc.
- Le rôle (*Role*) : il spécifie la manière dont un composant se comporte.
- Le type (*Kind*) : Il existe deux types de composants :
 - Les composants logiques : ce sont les paquetages liés à une fonctionnalité. Ils sont utilisés pour contrôler la complexité du système. *Domain Component* fournit les contrôles pour la gestion du projet, *Data component* fournit les accès aux bases de données, leur validation et leur conversion, et *User component* fournit les fonctionnalités d'interface utilisateur et les accès aux 2 autres composants.

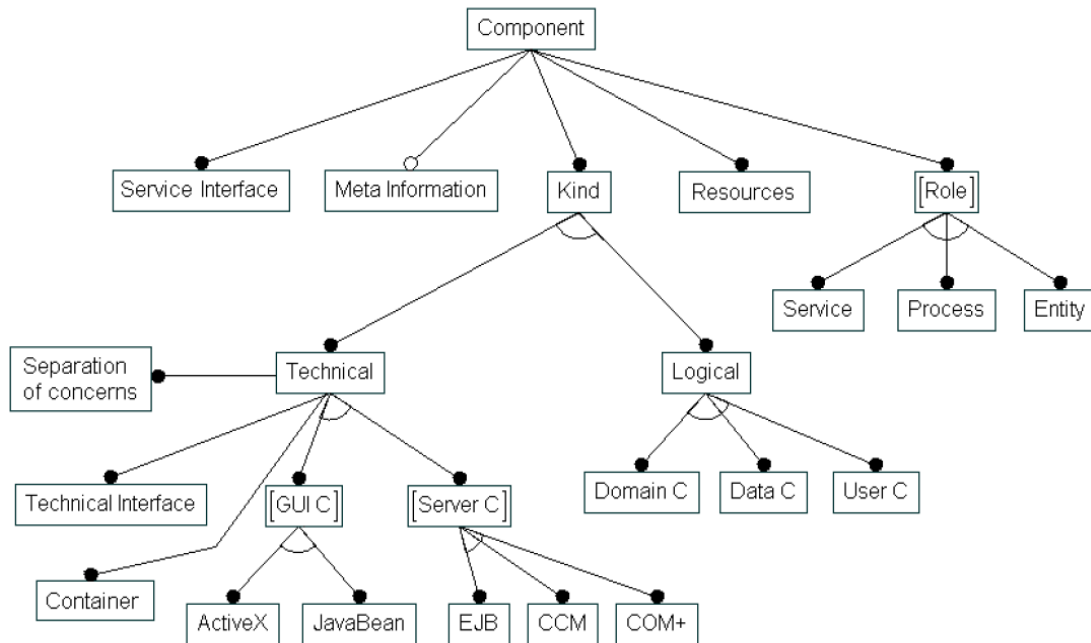


Figure 12 : Taxonomie des composants. [24]

- Les composants techniques : Ils représentent les briques techniques qui permettent d'assembler les applications logicielles. Ils sont composés d'un conteneur (*container*) qui fournit un environnement d'exécution pour le composant et permet d'évaluer et comprendre les ressources, les métas informations et les interfaces. Ils prennent en charge les préoccupations techniques pour le composant pour que le développeur ne les programme pas manuellement, d'où le concept *Separation of concerns*. Ces préoccupations peuvent être liées à la sécurité des composants, la fiabilité de fonctionnement, etc. L'interface Technique (*Technical Interface*) permet au conteneur d'accéder à tous les composants qu'il héberge pour les configurer, les arrêter, les démarrer, leur demander des informations, etc. Les composants techniques peuvent être utilisés dans les applications clients (*GUI component*), dans ce cas le conteneur est un environnement de développement intégré (IDE). Ou bien, ils peuvent être utilisés sur des tiers systèmes, dans ce cas, le conteneur se trouve dans les applications des serveurs utilisés.

3.3.3. Le logiciel durable

Le génie logiciel durable est l'approche d'ingénierie qui permet de définir et de développer un logiciel, en évaluant en permanence, en documentant et en optimisant ses effets négatifs et positifs sur la durabilité, qui résultent ou sont susceptibles de résulter dans son cycle de vie [25]. On se concentre principalement à ce niveau, sur la réduction de la consommation des ressources énergétique et matérielle dans tout le cycle de vie du logiciel et sur la réduction des déchets vers l'environnement. Ces objectifs peuvent se faire en optimisant les activités des phases du cycle de vie pour réduire leurs impacts sur l'environnement et en améliorant la qualité du logiciel. Dans ce dernier point, des métriques de qualité sont calculées pour nous renseigner sur la situation. On peut améliorer la qualité logicielle pour le développement durable par deux manières : en améliorant l'efficacité du logiciel par la réduction de sa consommation des ressources matérielles et ainsi on réduit sa consommation énergétique. Cela se fait par la programmation d'algorithme efficace, la réduction de la complexité, etc. La deuxième façon, traite la pérennité du logiciel et essaye de l'augmenter en améliorant les critères de qualité qui lui sont reliés comme la fiabilité, la convivialité, etc. L'optimisation des phases du cycle de vie peut se faire par différents moyens : réutilisation de code (ex. le modèle des composants), la virtualisation, la réduction des consommations des équipes du projet, etc.

Le Cycle de vie d'un logiciel [26] (voir figure 13) commence par le développement qui comprend la phase d'analyse et de détermination des besoins (*dans laquelle les besoins du client sont traduits*



Figure 13 : Cycle de vie du logiciel et ses impacts.

dans un cahier des charges), la phase de conception (où l'architecture du système est déterminée selon un modèle), la phase d'implémentation (dans laquelle le modèle de conception est traduit en code informatique), et enfin les tests qui permettent de vérifier et valider le logiciel. Après, la fin de la phase de développement, le logiciel est distribué au client, avec des activités de commercialisation, de documentation, et d'installation au niveau des infrastructures du client. L'utilisation commence après, suivie par de la maintenance. En fin de vie du logiciel, L'utilisation du logiciel s'arrête, il est désinstallé, ses données sont converties et sauvegardées. Le code source est compressé et archivé aussi pour des futures réutilisations.

Chacune de ces phases a 3 différents impacts sur l'environnement [27] [28]:

- Impacts de premier ordre : ils comprennent tous les impacts directs des phases du cycle de vie du logiciel sur l'environnement et qui résultent des consommations énergétiques et de matières, et le rejet des déchets.
- Impacts de second ordre : ce sont des impacts positifs, ils se focalisent sur les valeurs ajoutées par les techniques de l'informatique utilisées pendant le cycle de vie du logiciel, dans la société et l'économie, par exemple les avantages de la virtualisation qui permet de réduire le nombre de ressources matérielles, la dématérialisation qui permet de substituer les biens tangibles par du capital immatériel et des services (exemple : acheter un logiciel sur internet en le téléchargeant, au lieu de l'acheter sur un support CD), etc.

- Impacts de troisième ordre : regroupe tous les effets indirects des phases du cycle de vie du logiciel. Le logiciel altère d'autres services, processus et produit qui lui sont externes. Ces changements produisent des impacts de troisième ordre. Par exemple la réservation dans un site web de voyage a des effets indirects sur les services de transports.

Il existe plusieurs métriques (voir tableau 2) pour mesurer la durabilité du logiciel (consommation énergétique, rejet de CO₂, Besoin en ressources matérielles, etc.). Ces métriques principalement, traduisent l'utilisation des ressources matérielles par le logiciel en impacts. Dans une étude d'analyse pour recenser tous les métriques de durabilité logicielle qui existent, dans la littérature scientifique, [29] ont trouvé qu'il y a 93 métriques qui mesurent la durabilité logicielle, 66 d'entre elles concernent la consommation énergétique du logiciel.

Metrics Type	Total	Measurement Unit(s)
Energy	48	Joule (J), Index, Watt (W), Ampere (A) Kilowatt-hour (kWh), Number, byte/kWh
Performance	19	GFLOPS/kWh, Computing Unit/kWh, Percentage (%), Seconds (s), Index, Number
Utilization	17	Percentage (%), Megabyte (MB), Megahertz (MHz), GB/s
Economic	9	Dollars (\$)
Performance / Energy	2	GFLOPS/Watt, Index
Pollution	1	CO ₂ units

Tableau 2: les types de métrique de durabilité du logicielle. [29]

Parmi les unités proposées dans le tableau 2, certaines sont dénuées de sens physique et ne peuvent pas être utilisées dans un travail d'ingénierie. Par exemple l'ampère (A) est associé à la mesure de l'énergie, alors que réellement, l'ampérage est utilisé pour mesurer l'intensité d'un courant électrique et non pas l'énergie.

3.4. Synthèse du chapitre

Les robots sont des entités issues de l'intégration synergétique d'une partie logicielle dans une partie matérielle. La première encapsule ses connaissances et ses comportements, afin de contrôler les éléments constituant sa partie matérielle, lui permettre de donner un sens à sa perception et de produire des décisions. La partie matérielle lui permet d'exécuter la partie logicielle et de traduire ces décisions en actions physiques dans l'environnement. Ces deux parties sont donc complémentaires et indissociables.

En outre, Le rapport entre le robot et son environnement est existentiel ; le robot est construit à partir de ressources issues de l'environnement et fonctionne selon la perception des changements physiques intervenus dans son milieu, il prend des décisions et agit dans l'environnement. Il est évident que le robot doit être associé à un environnement. Enfin, la granularité de la relation du robot avec son environnement est définie par les exigences du client, les choix de conception et les flux énergétiques et matériels du robot. Afin de développer des robots durables, Il faut optimiser les performances environnementales de ses deux parties logicielle et matérielle, et analyser leurs impacts dans tous leurs cycles de vie pour les réduire.

Les notions présentées dans ce chapitre vont nous servir dans la construction de notre méta modèle et nous permettre de spécifier ses concepts ainsi que leurs propriétés.

Chapitre 4

Le méta modèle de la robotique durable

Ce chapitre présente le méta modèle « SuRo » (acronyme de *Sustainable Robotics*) que nous proposons pour modéliser le domaine de la robotique durable. Nous nous sommes fondés sur les principes de l'ingénierie dirigée par les modèles pour créer notre modélisation. Elle est basée sur les hypothèses définies dans l'introduction : le robot est modélisé selon ses deux parties logicielle et matérielle, leurs interactions avec l'environnement sont prises en considération, dans une perspective de modélisation des impacts et des échanges de flux avec l'environnement. Ce dernier est un concept essentiel dans notre modèle. SuRo Intègre principalement les notions que nous avons définies dans notre analyse.

4.1. L'ingénierie dirigée par les modèles

L'ingénierie dirigée par les modèles en anglais : *Model Driven Engineering* (IDM), permet de maîtriser la complexité liée au développement logiciel, par des analyses et des transformations à plusieurs niveaux des modèles, qui abstraient le problème (ou le système) étudié, en une implémentation logicielle générée automatiquement. Cette discipline permet de se déplacer d'une ingénierie logicielle classique, où tout est centré sur les objets, vers une nouvelle approche de développement, où tout est centré sur les modèles.

Un modèle est une abstraction d'un système, modélisé sous la forme d'un ensemble de faits construits dans une intention particulière [30]. Le modèle représente le système étudié (*System under study*) en le simplifiant pour nous permettre de comprendre, prédire et maîtriser ses états, sa structure, ses fonctionnalités, etc. Il doit être conforme à un méta modèle, ce dernier définit les notions essentielles qui décrivent le modèle. Principalement, il est associé à un langage de méta-modélisation composé d'une syntaxe et d'une sémantique. La figure suivante décrit les relations qu'a un modèle avec le système et le méta modèle.

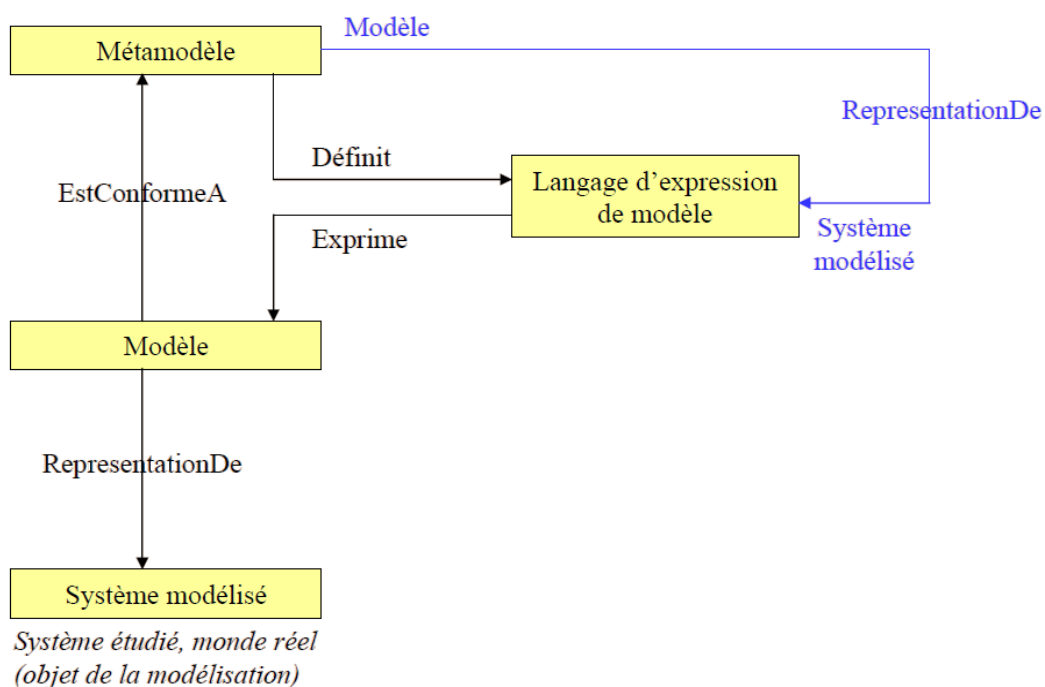


Figure 14 : Relations entre système, modèle et Meta modèle. [40]

Afin de standardiser et d'organiser les concepts de l'ingénierie dirigée par les modèles, l'OMG a défini l'architecture dirigée par les modèles (MDA). L'objectif de cette architecture est de réunir tous les avantages offerts par l'IDM tels que la pérennité, la productivité, la prise en compte des plateformes d'exécution, etc. Pour les exploiter et prescrire les bonnes pratiques de la modélisation [30]. L'architecture MDA est sous forme d'une pyramide (voir la figure 15) constituée des quatre niveaux suivants [31] [32] :

- **Le niveau M0** : il représente le système à étudier (*System under study*), généralement c'est le monde réel avec les informations, les comportements, les états et les interactions qu'on souhaite modéliser.
- **Le niveau M1** : il est composé des modèles d'informations qui permettent de décrire les informations du M0, UML par exemple fait partie des modèles de ce niveau. Le M1 regroupe tous les modèles qui sont exprimés dans un langage unique dont la définition est fournie explicitement au niveau M2.
- **Le niveau M2** : il définit le méta modèle dont le modèle appartenant au niveau M1 doit être conforme. le méta-modèle est associé à un langage spécifique à un domaine, qui est composé d'une syntaxe et d'une sémantique uniques ou à un profil qui définit une variante de ce langage.
- **Le niveau M3** : Il définit le méta-méta modèle par rapport auquel le méta modèle du niveau M2 doit être conforme. Généralement, ce méta-méta modèle est appelé un MOF (*Meta-Object Facility*). Le MOF décrit la structure et les fonctionnalités universelles de tous les méta-modèles qui se trouvent au niveau M2. il est réflexif, c'est-à-dire que la description du MOF s'applique au MOF lui-même, ce qui lui permet d'être le dernier niveau de l'architecture MDA.

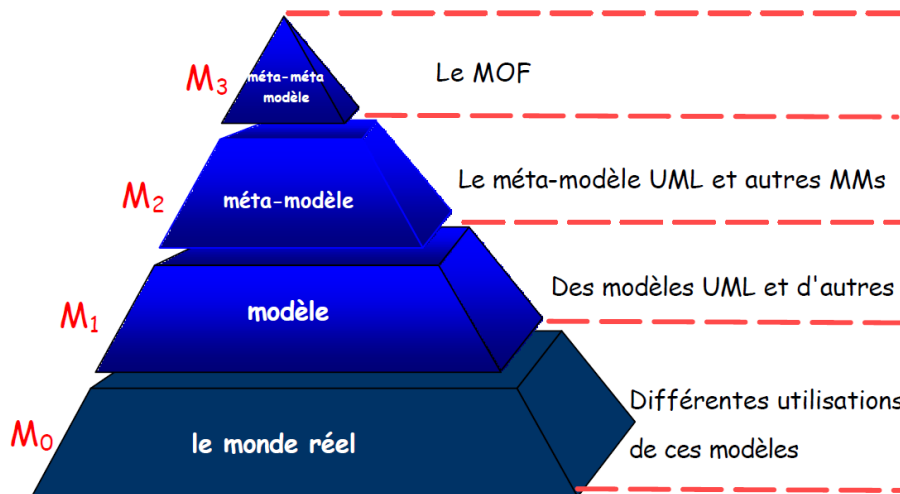


Figure 15 : L'architecture MDA d'OMG. [31]

En pratique l'architecture MDA admet principalement les modèles suivants [33]:

- Le CIM (*Computation Independent Model*) : il représente la vision métier ou le domaine d'application du système étudié, en tenant compte de son environnement et indépendamment de toute plateforme informatique. Le CIM abstrait la structure du système étudié et permet de représenter uniquement ses fonctionnalités. Son indépendance technique lui assure une pérennité au cours du temps, puisque il est modifié seulement lorsque les connaissances ou les besoins métier changent.
- Le PIM (*Plateforme Independant Model*) : il représente la logique métier spécifique au système ou son modèle de conception, indépendamment de toute plate-forme technique. il décrit une

vue partielle d'un CIM, par exemple le fonctionnement des entités, des services, les transactions, etc.

- Le PSM (*Plateforme Specific Model*) : il est dépendant de la plate-forme technique et il décrit comment le système l'utilisera. Il fournit principalement les informations qui permettent de générer le code informatique, exécutable sur cette plateforme.
- Le PDM (*Platform Description Model*) : il contient les informations de transformation qui assurent le passage du PIM au PSM.

Le passage du monde réel (*System under study*), vers la génération du code informatique se fait selon la succession de plusieurs transformations (voir la figure 16), décrites comme suit [31] [33] :

- *Transformation de PIM vers PIM* : ce passage permet de raffiner le model PIM en le modifiant pour l'enrichir, le filtrer ou le spécialiser. Ce raffinement se fait par l'ajout ou la soustraction d'informations fonctionnelles.
- *Transformation de PIM vers PSM* : ce passage permet de spécialiser le PIM raffiné selon les caractéristiques d'une plateforme technique. Cette transformation en PSM se fait en ajoutant au PIM des informations sur la plateforme technique (CORBA, XML, etc.) qui permettent de générer le code informatique. Ces caractéristiques de transformations sont fournies par le PDM ou le MOF.
- *Transformation de PSM vers PSM* : elle s'effectue lorsque le passage d'un PIM vers un PSM n'est pas suffisant pour générer le code informatique. Cette transformation se déroule lors des phases de déploiement, d'optimisation ou de reconfiguration, en utilisant des formalismes intermédiaires.
- *Transformation PSM vers PIM* : cette transformation permet le passage du code ou du PSM existants vers un modèle PIM, en appliquant les notions de la rétro-ingénierie (reverse engineering).

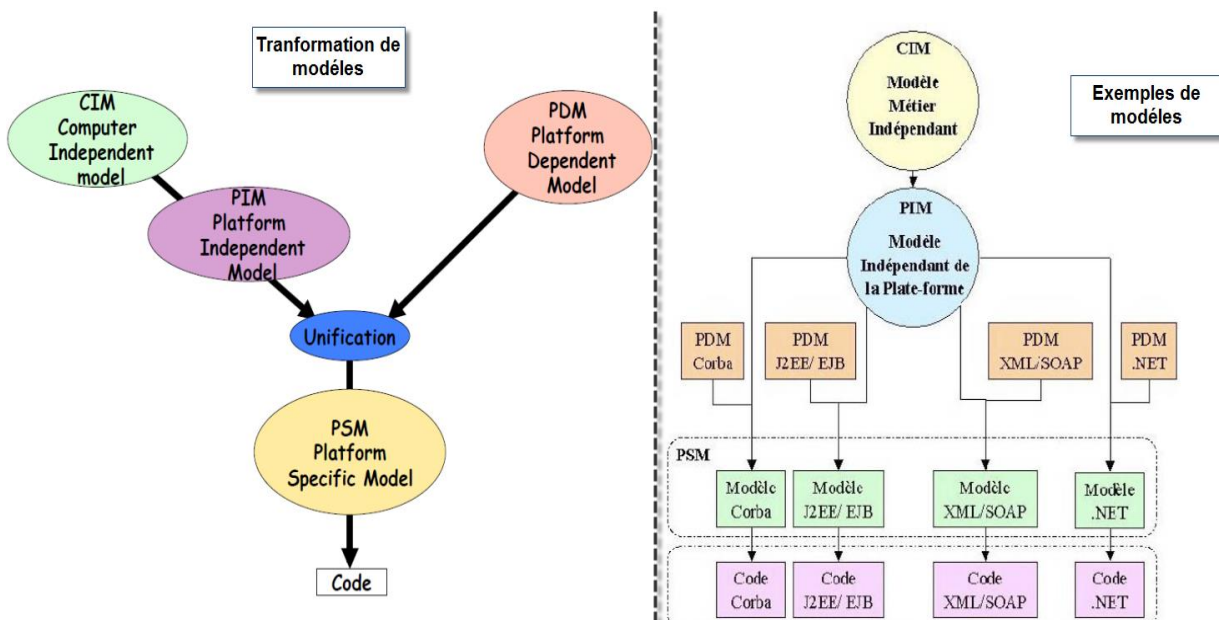


Figure 16 : Transformation et exemple de modèles en MDA. [41]

En robotique (voir la figure 17) [34], généralement le passage du PSM vers le code s'effectue tout d'abord par une transformation du modèle PSM vers un modèle de code spécialisé sur une plateforme spécifique d'implémentation (PSI). Ce passage s'effectue en ajoutant par exemple le code utilisateur, les bibliothèques, etc. Ensuite, le modèle de code est transformé en un code exécutable, qui sera déployé dans le robot.

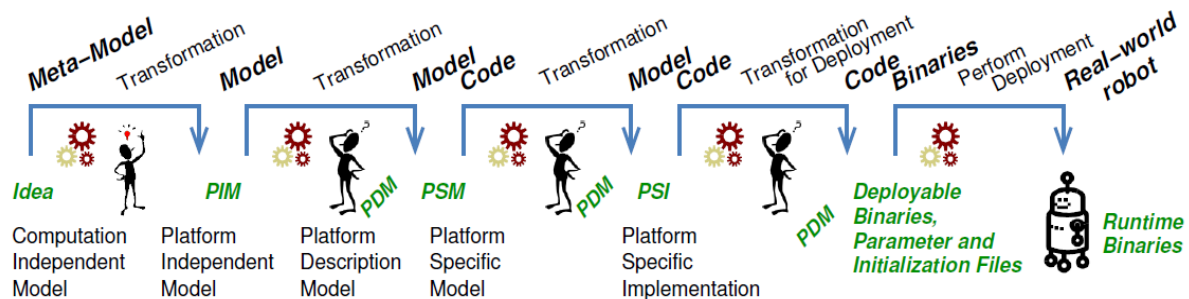


Figure 17 : Transformations MDA en robotique. [42]

4.2. Le CIM de SuRo

Dans notre travail, nous nous préoccupons d'une conception de robot dirigée par les modèles, qui permet de simuler le domaine de la robotique durable, en se basant sur les notions de la MDA. SuRo peut être enrichi, transformé et déployé pour produire une simulation en offline, basée sur plusieurs scénarios. Ces derniers sont liés aux choix des concepteurs et aux contraintes spécifiées dans l'analyse des besoins. Chaque simulation représentera une instantiation de SuRo selon les informations qui lui sont ajoutées. Elle permettra de vérifier et prédire les impacts des choix sur l'environnement, et de valider le meilleur scénario. En d'autre terme, le méta modèle permettra aux parties prenantes de prendre les décisions les plus adéquates ; celles qui permettent une meilleures durabilité du robot. SuRo est d'une part une réponse à la complexité de la robotique durable qui est basée sur l'intégration des notions du développement durable dans le domaine de la robotique (Génie logiciel, électronique et mécanique) et qui nécessite des experts polyvalents qui doivent maîtriser tous ces domaines, et d'une autre part, une première contribution dans ce domaine pour ouvrir la voie à des futures recherches scientifiques. Les objectifs que vise SuRo sont :

- Soutenir, promouvoir et mettre en lumière la synergie inhérente qui existe entre le robot et l'environnement, dans une optique où tout vient de l'environnement et repart dans l'environnement.
- Réduire les impacts sur l'environnement, les coûts énergétiques liés à la consommation et se placer dans une perspective dans laquelle la compensation et les directives environnementales seront obligatoires dans le future.
- Faciliter la réutilisation, la portabilité, la productivité et la pérennité en se basant sur les notions de l'IDM.
- Permettre l'expérimentation de plusieurs scénarios, par des simulations générées rapidement et sans effort. Afin de maîtriser la complexité du domaine.
- Valider les hypothèses sur lesquelles se base notre travail.

L'univers dans SuRo est un super-système composé d'une imbrication de plusieurs systèmes qui sont en interactions. Chaque système est composé d'un référentiel temps représentant la référence qui permet de capturer l'évolution du système, d'un environnement représentant l'espace physique du système avec tous ce qui le compose, et d'entités. L'entité est un concept central au système. Elle incorpore une partie informationnelle dans une partie matérielle, dotée de comportements qui lui permettent d'agir sur l'environnement, d'effectuer des interactions avec d'autres entités et les éléments de l'environnement. Le robot est un type d'entité, comme on l'a mentionné plus haut dans ce rapport, il incorpore une partie logicielle (informationnelle et intangible) dans une structure

matérielle et possède plusieurs comportements qui lui permettent d'interagir avec l'environnement et avec d'autres entités. Un autre type d'entité peut être l'être humain, un groupe d'êtres humains, une organisation d'êtres humains (ex : parties prenantes), etc. Selon le contexte on peut avoir plusieurs types d'entités qui peuvent être mis en lumière. L'environnement est constitué d'un espace et de ressources. Le premier concept décrit la frontière physique du système en termes de distance, de localisation et de géométrie. Le concept de ressources représente tous notions qui obéissent aux lois physiques et qui sont utilisées, transformées, recyclées, réutilisées, et consommées par les entités. La figure 18 décrit l'idée générale de SuRo. Il est composé principalement des concepts : univers, système, environnement, entité (dans le schéma le robot et les parties prenantes) et temps.

Le robot est décrit par l'axe vertical qui est composé en haut du concept structure matérielle et en bas par le concept structure logicielle. Chacun de ces concepts a un cycle de vie composée de plusieurs phases, des durées pour chaque phase et une durée de vie totale. Cela indique la relation avec l'axe temps. Les parties prenantes interviennent dans toutes les phases des cycles de vie de ces deux concepts. Ces acteurs déterminent par leurs choix, leurs contraintes et leurs objectifs, les concepts : structure matérielle, structure logicielle, et leurs interactions avec le temps et l'environnement.

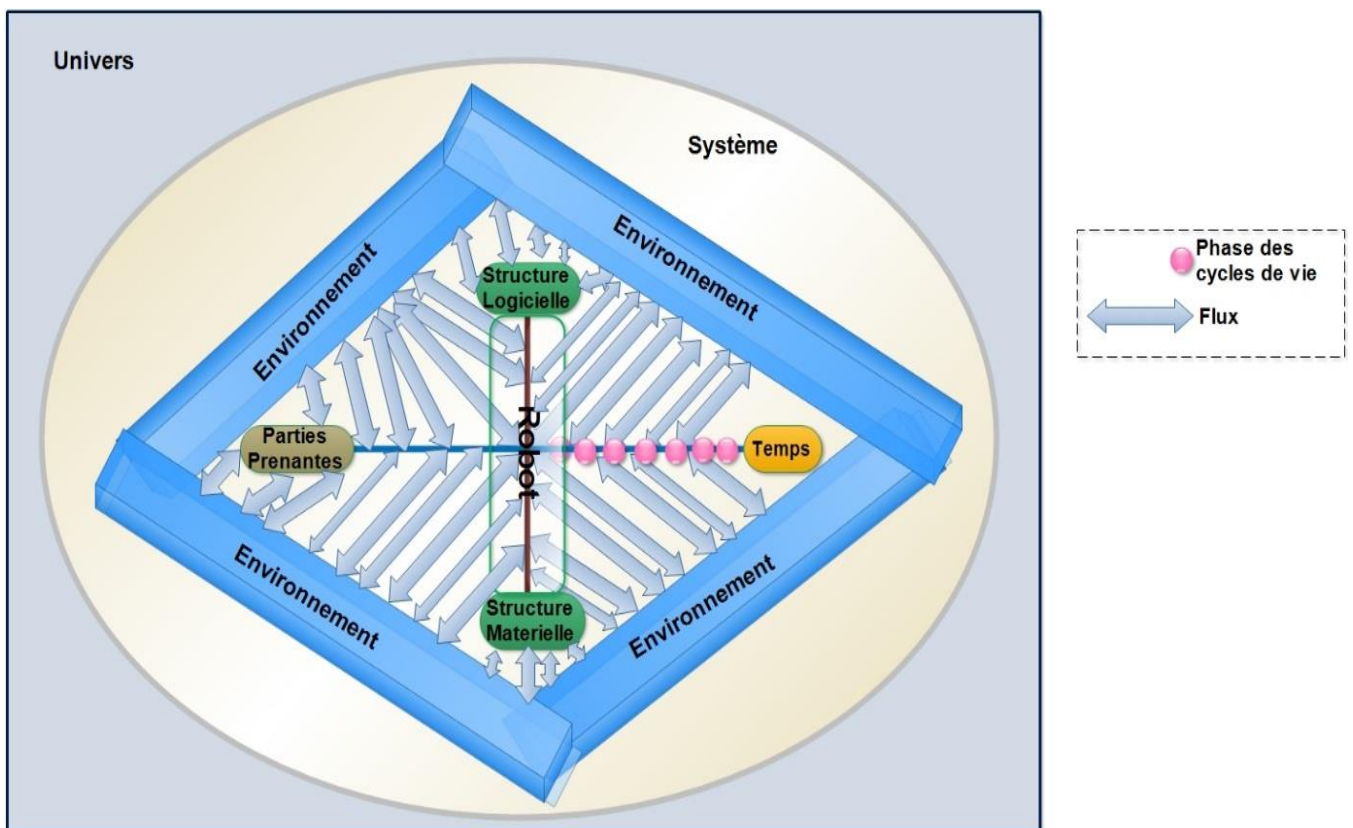


Figure 18 : Idée principale de Suro.

Les concepts de structure matérielle et de structure logicielle échangent des flux de ressources avec l'environnement durant leurs phases de vie dans lesquelles interviennent les parties prenantes. Ces dernières échangent aussi des flux avec l'environnement durant ces phases. Les flux peuvent être des flux d'entrée, reçus par l'environnement ou bien des flux de sortie transmis par ces derniers vers l'environnement.

Un point essentiel dans l'idée de SuRo est la pérennité de l'environnement. Celui-ci existe avant le robot, et continue d'exister après la fin de vie du robot.

Tous ces concepts font partie d'un système qui décrit dans notre cas, le domaine de la robotique durable. On peut avoir plusieurs systèmes qui interagissent. Ces derniers forment un tout qui est l'univers.

La figure 19 décrit le modèle computationnel indépendant (CIM) de SuRo. Le modèle est composé de plusieurs concepts qui mettent en évidence l'idée principale décrite plus haut. Le tableau suivant définit explicitement ces concepts :

Concept	Description
Universe	La totalité de tous les systèmes qui existent et leurs interactions.
System	L'assemblage de toutes les entités et leurs interactions avec le temps dans l'environnement.
Time	La mesure du changement associée au passage d'un état à un autre.
Environment	Le milieu où les entités évoluent et avec qui elle échangent des flux.
Entity	Le concept central du système qui incorpore une partie informationnelle dans une partie matérielle pour la doter de comportements.
Phase	Une étape qui caractérise un intervalle de temps.
Process	Une fraction d'une phase qui caractérise une évolution d'un état à un autre.
Space	Une zone à trois dimensions dans laquelle les éléments du système sont localisés.
Ressource	Moyens physiques disponibles dont l'entité a besoin pour exister.
Energy	La source de puissance qui permet à l'entité d'être active.
RawMaterial	La matière première substantielle qui est utilisée comme source pour construire la partie matérielle de l'entité.
Stakeholder	Les parties prenantes (acteurs) qui interviennent dans toutes les phases de vies du robot et qui l'affectent avec leurs décisions et leurs actions.
Component	Un élément de l'environnement doté de comportements qui lui permettent d'agir au sein du système.
Robot	Une entité créée, composée d'une partie logicielle incorporée dans une structure matérielle capable de comportements.
Cost	Les coûts en temps, en effort et en argent des entités dans leurs phases de vie.
Flux	Les quantités de ressources échangées entre l'entité et l'environnement.
Behaviour	Les fonctionnalités de l'entité qui lui permettent d'interagir dans le monde.
Task	Les actions d'une entité dans le temps
Structure	Une organisation des parties matérielles et informationnelles qui constituent l'entité.
MaterialPart	Les éléments tangibles de l'entité qui obéissent aux lois de la physique.
InformationPart	Les éléments incorporels de l'entité qui lui permettent d'abstraire le monde.
Software	Une partie informationnelle sous forme d'un programme informatique qui abstrait les comportements des entités.
Knowledge	Une connaissance acquise dans le temps par expérience, déduction, apprentissage, etc. et qui permet à l'entité de donner un sens aux éléments du système.
Quality	Une caractéristique ou propriété attendu par le stakeholder et qui permet de donner une valeur d'estime à l'entité ou ses parties.
Data	Un renseignement cohérent, intègre et consistant, qui peut être fixe ou variable.
Impact	Les mesures des effets des comportements des entités dans chaque phase du cycle de vie, caractérisant les flux échangés avec l'environnement.

Tableau 3: la définition des concepts de SuRo.

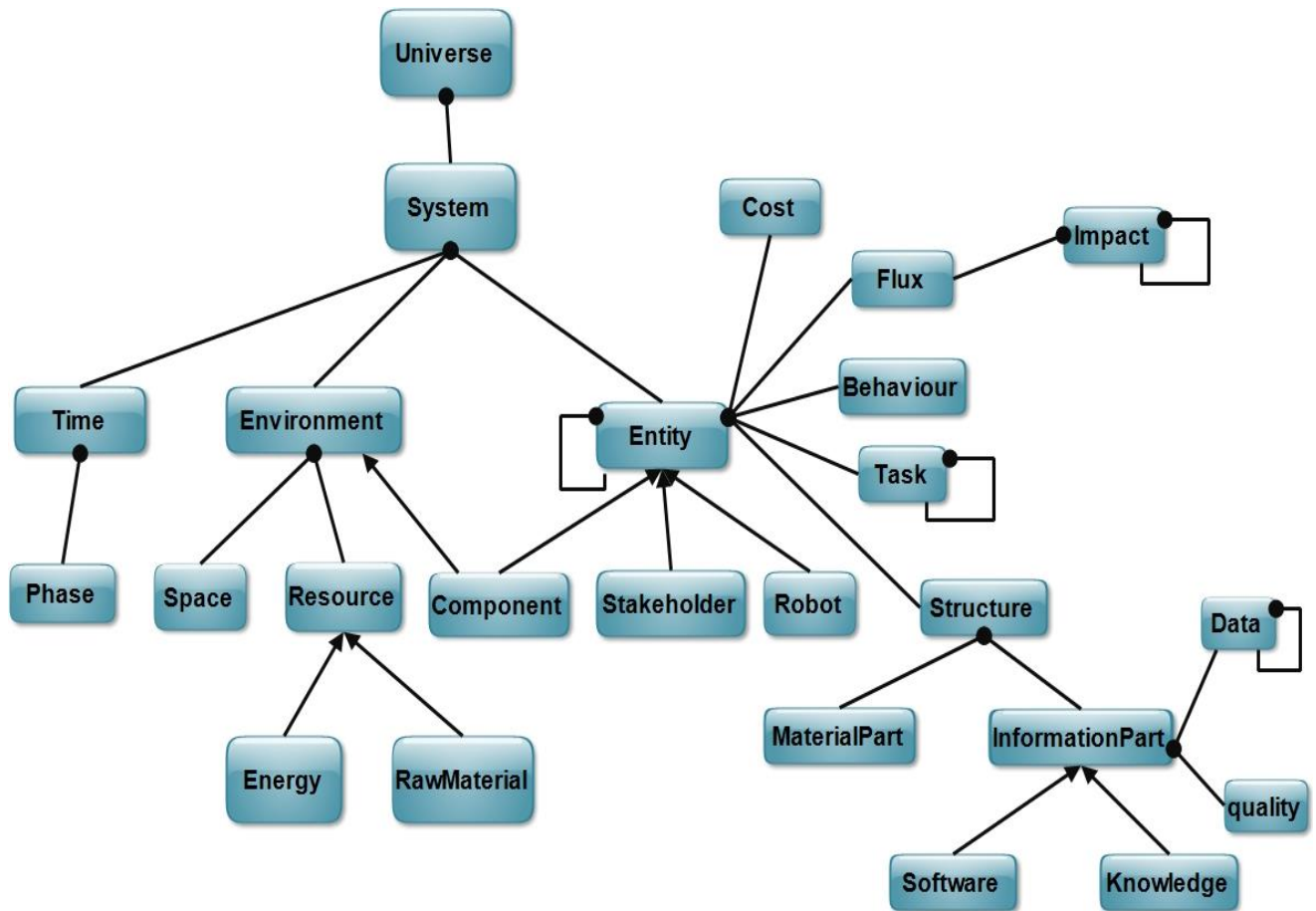


Figure 19 : Le CIM de Suro.

Dans ce modèle les concepts sont liés par des relations de composition (ligne avec cercle) et des relations de spécialisation (ligne en forme de flèche). Les autres types de relations qui peuvent exister, seront explicités dans le PIM du méta modèle. Ces dépendances sont :

- Un univers est composé d'au moins d'un système.
- Un système est composé d'une référence de temps, d'un environnement et d'au moins une entité.
- Le temps est composé de plusieurs phases.
- L'environnement est composé d'un espace et de plusieurs ressources.
- Chaque ressource peut être sous forme d'une énergie ou de matières premières.
- Une entité peut contenir d'autres entités.
- Une entité peut être un robot, un composant ou un stakeholder.
- Une entité possède un ou plusieurs coûts. Elle échange un ou plusieurs flux. Elle possède un ou plusieurs comportements, effectue une ou plusieurs tâches et est composée d'au moins d'une structure.
- Chaque flux a un ou plusieurs impacts.
- Un impact peut générer un ou plusieurs autres impacts.
- Une tâche peut contenir une ou plusieurs autres tâches.
- Une structure possède au moins une partie matérielle et une partie information.
- La partie information peut être un logiciel ou bien des connaissances.
- La partie information possède des données et des qualités.
- Une donnée possède une ou plusieurs informations.

4.3. Le PIM de SuRo

Nous avons utilisé le méta modèle et le langage Ecore proposés par EMF (*Eclipse Modeling Framework*) pour modéliser le PIM de SuRo. EMF est une Framework de modélisation intégrée dans Eclipse. Elle est basée sur les principes de l'IDM et suit l'architecture MDA. Elle permet de générer automatiquement le code en se basant sur un modèle de données structuré. EMF fournit des outils et un support d'exécution pour produire un ensemble de classes Java pour le modèle, un ensemble de classes qui permettent la visualisation du modèle, un éditeur de base. Elle fournit également, les bases pour l'interopérabilité avec d'autres outils et applications basées sur EMF [35]. Ecore est un outil intégré dans EMF qui offre un méta modèle pour la conception d'architecture MDA. Il offre un langage pour créer d'autres langages et des métas modèles en se basant sur les notions d'UML. Nous avons utilisé cet environnement de modélisation pour les raisons suivantes :

- La robustesse des modèles par rapport à leurs métas modèles Ecore, puisque la génération du squelette du code assure la conformité au méta modèle.
- La simplicité d'utilisation et la convivialité de l'interface.
- L'extensibilité qu'il offre, par exemple il est possible d'ajouter des plugins pour améliorer ou adapter le méta modèle.
- Les possibilités d'interopérabilité avec tous les métas modèles créés en utilisant la plateforme EMF.

Les concepts principaux que fournit Ecore [36] [37] sont définis comme suit (voir figure 20):

- EClass : il représente le modèle de classes. Les classes sont identifiées par leur nom et leurs caractéristiques (les attributs et les références). Une classe peut hériter d'un certain nombre d'autres classes (super-classes). Une classe peut être abstraite, dans ce cas, elle ne peut pas être instanciée. Ou bien, sous forme d'une interface dans ce cas, aucune implémentation ne lui sera générée).
- EPackage : le modèle de package est un conteneur qui rassemble plusieurs classe et les types de données.
- EFactory : le modèle d'usine permet d'instancier les classes et de convertir les valeurs des données en chaîne de caractères et vice versa.
- EAnnotation : le modèle des annotations permet d'associer et de décrire n'importe quel autre élément du modèle par des informations supplémentaires.
- EAttribute : le modèle d'attribut permet de définir les propriétés des éléments du model.
- EDataType : le modèle des types de données. Il permet de spécifier des types de données quelconques.
- EOperation : le modèle des opérations. Elle permet de décrire les opérations qui peuvent être invoquées dans une classe quelconque.
- EParameter : le modèle des paramètres. Il spécifie les types des paramètres qui peuvent être passés comme arguments dans n'importe quelle opération.
- EEnumLiteral : il représente les membres d'un ensemble de type d'énumération qui ont des valeurs littérales.
- EInheritance : il représente la relation d'héritage entre classes. Elle est représentée par une flèche avec une tête triangulaire vide.
- EReference : il représente l'association entre deux classes. Il peut être une relation sans confinement, dans ce cas, la relation est représentée par une flèche noire. Ou bien, il peut être avec confinement, représenté par une flèche à tête de losange. Une référence est accompagnée par une cardinalité dotée d'une limite inférieure et d'une limite maximale.

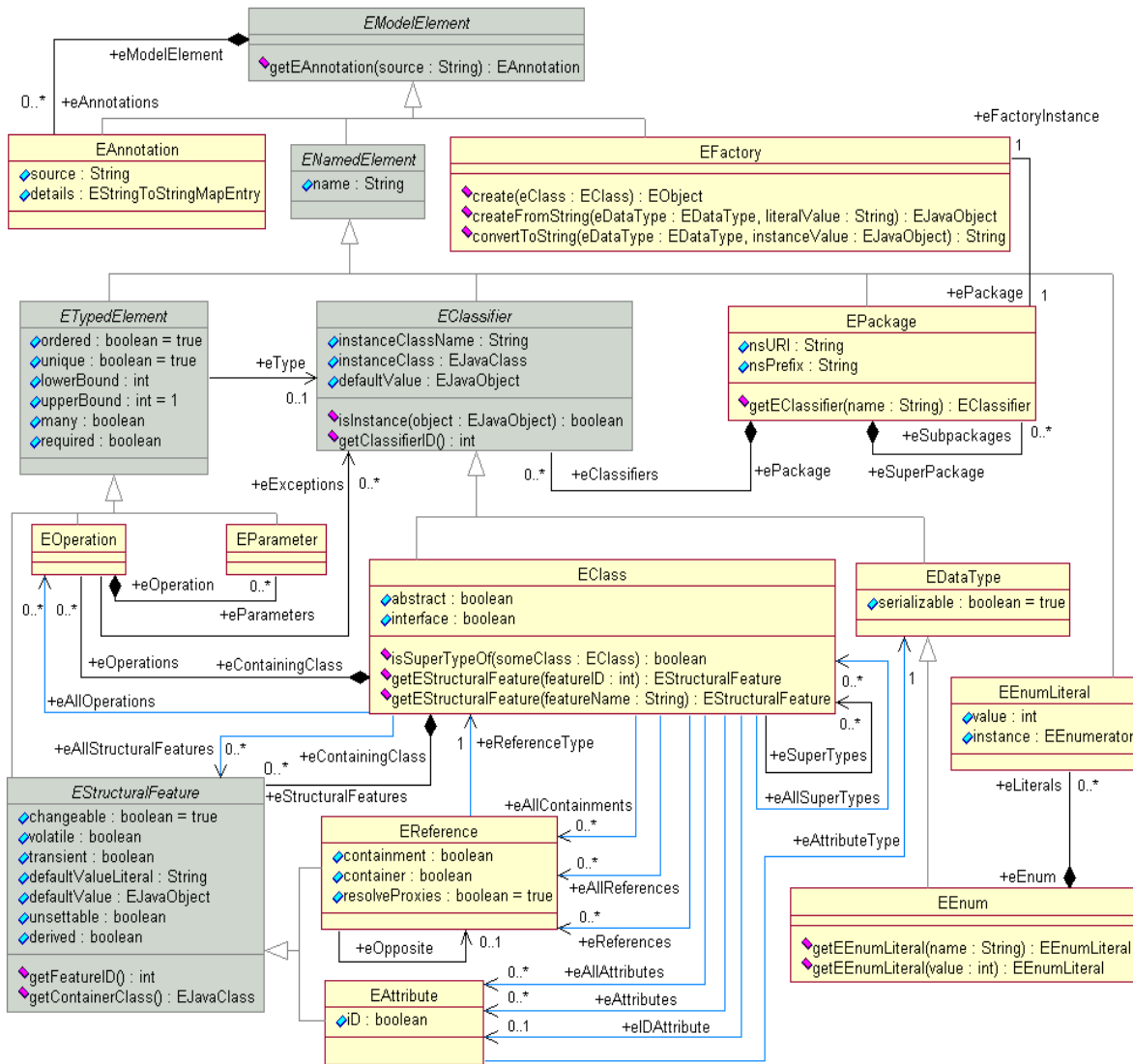


Figure 20 : Le diagramme de classe du méta-méta modèle d'Ecore. [43]

La figure 21 représente le PIM de base de notre méta modèle, tels que définie plus haut. Les éléments qu'on a ajoutés sont comme suit :

- Des énumérations pour exprimer les littéraux associés aux attributs:
 - FluxType : pour décrire les types de flux selon la description qu'on a fournie dans la partie analyse du rapport.
 - ScopeType : Le type de l'environnement de l'entité, il peut être naturelle, fait par l'humain (maison, usine, etc) ou bien les deux à la fois.
 - PhysicalUnit : elle représente l'unité physique qui permet de quantifier les variables physiques.
 - SourceType : il représente la source d'où provient l'énergie.
 - MaterialType : il décrit le type de matière dans l'environnement (Biotique pour la matière vivante et abiotique pour celle qui est non vivante).
 - ImpactType : il décrit le type d'impact selon notre analyse définie dans le chapitre 3.
- Des relations entre les concepts pour exprimer leurs interactions :
 - Express : Un logiciel exprime un ou plusieurs comportements d'entités.
 - In: un flux prend des quantités de ressource de l'environnement dans un espace.
 - Out : un flux rejette des quantités dans l'environnement dans un espace.
 - EvolveWith : les entités évoluent dans le temps selon des phases.

Le deuxième niveau de modélisation (voir figure 22) étend le concept *MaterialPart* en l'enrichissant avec les notions décrites dans le chapitre 3 du rapport.

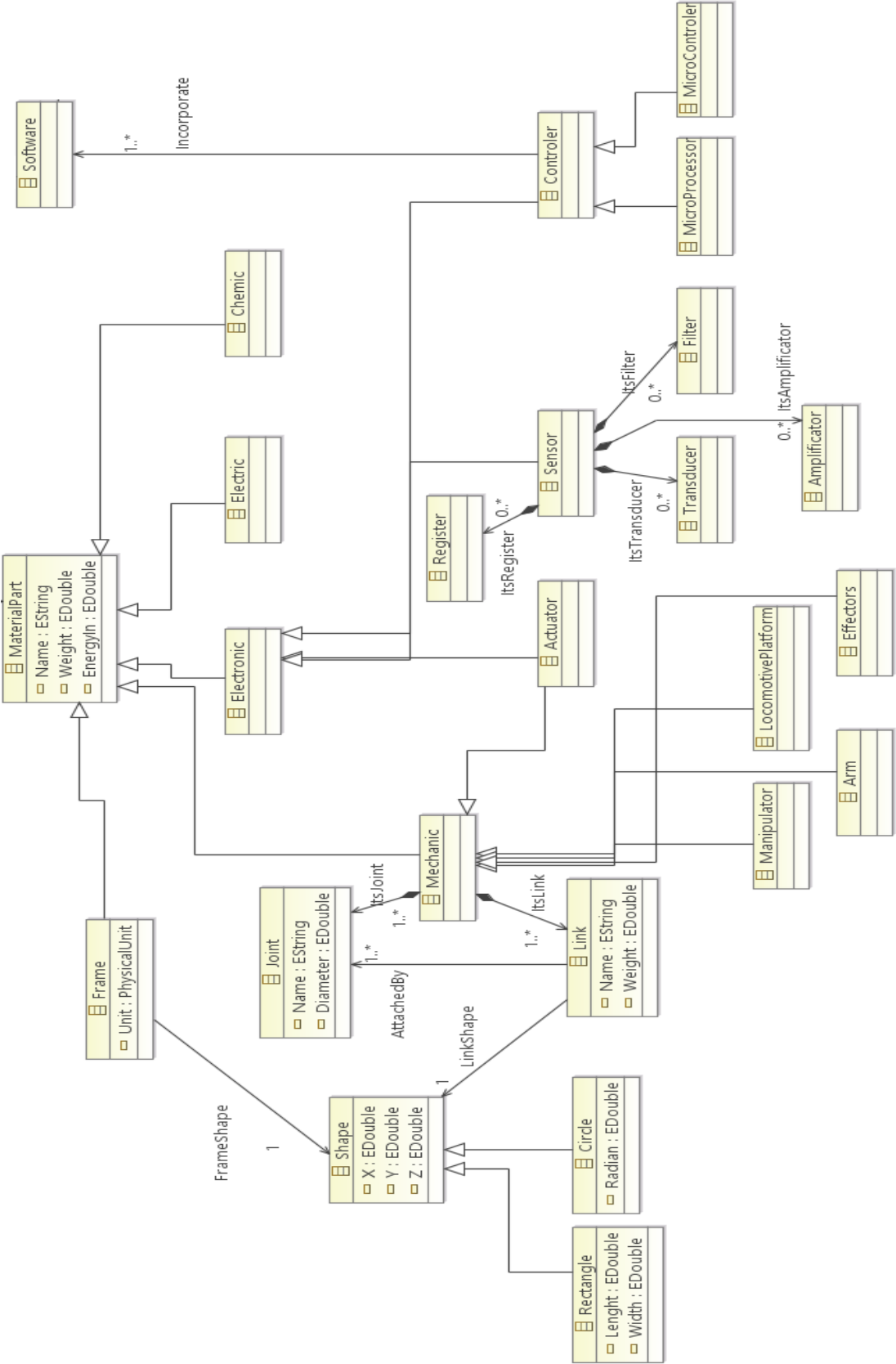


Figure 22 : Le niveau 2 de SuRo.

Ce niveau décrit la partie matérielle du robot ; *Frame* fait allusion au châssis. Celui-là à une *Shape* (forme) qui peut être sous différentes configurations. Les corps mécaniques (*Link*) ont aussi une *Shape* et ils sont attachés par des jonctions (*Joint*). L'actionneur (*Actuator*) hérite des concepts *Mechanic* et *Electronic*, puisqu'il est un dispositif électromécanique. Et enfin, un contrôleur (*Controler*) doit impérativement incorporer un logiciel. Vous remarquez aussi que nous avons ajouté deux propriétés particulières dans la classe mère *MaterialPart* ; *Weight* pour mesurer le poids des différentes parties matérielles et *EnergyIn* pour estimer leurs consommations énergétiques.

Le troisième niveau de SuRo comme indiqué dans la figure 24, indique les concepts de *Software* selon la taxonomie de [24], qui a été décrite dans le chapitre 3. Nous avons aussi, ajouté un attribut *EnergyIn* pour mesurer la consommation énergétique de chaque composant logiciel.

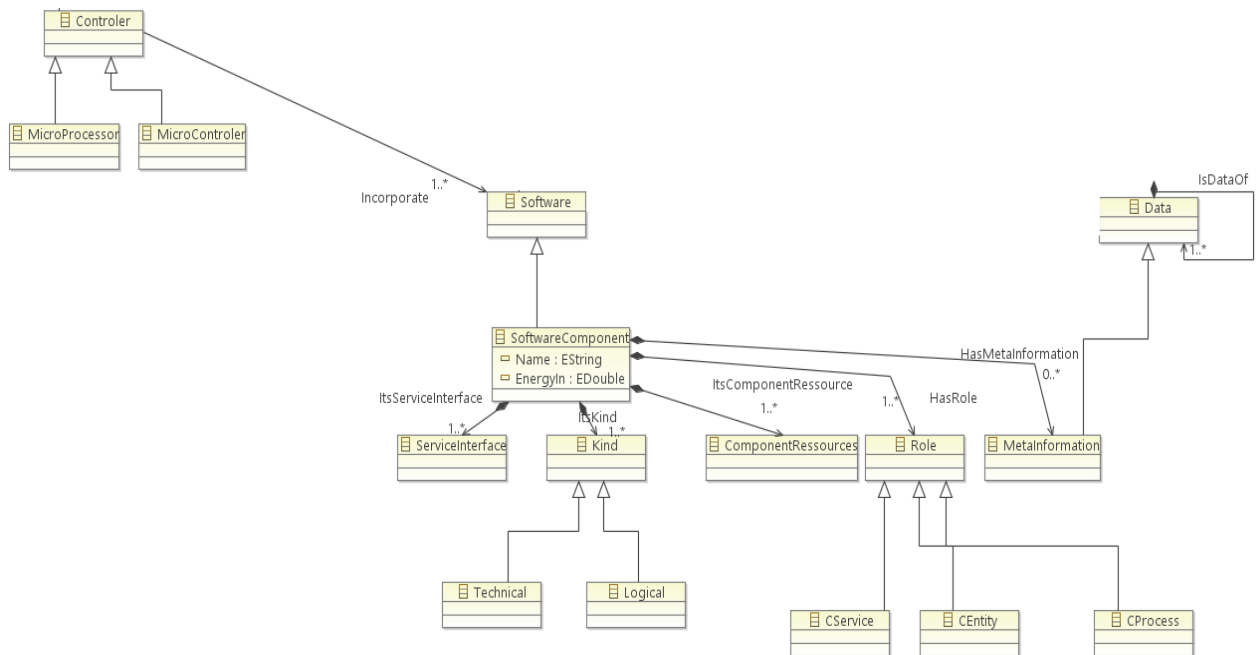


Figure 23 : Le niveau 3 de SuRo

4.4. Le PSM de SuRo

Le squelette du code source de SuRo peut être généré en quelques instants en utilisant le générateur de modèle de code d'Ecore. Le langage de programmation utilisé est java. Ecore offre aussi la possibilité d'inclure avec le projet du modèle code la documentation du développement, ou pour l'utilisateur. Une autre option permet d'instancier toutes les classes, leurs attributs et offre même la possibilité de tester le code en l'instanciant.

4.5. Liens avec la méthode de développement en X

La méthode de développement en X, spécifiée par [38], peut être considérée comme un CIM abstrait de SuRo. Cette méthode (voir la figure 24) définit les processus de développement du robot en prenant en considération la partie matérielle, la partie logicielle, leurs intégrations et leurs interactions avec l'environnement, depuis la formalisation de l'idée jusqu'à la fin de vie du robot. Les points communs de Suro avec cette méthode sont :

- 1- L'approche en X décompose les processus de développement en explicitant deux couches : une couche supérieure qui représente la partie immatérielle du système (idées, informations, logiciel, etc.) et dont les flux ont des conséquences indirectes sur l'environnement, et une couche

inférieure qui définit les composants matériels du système et ayant des interactions et des effets directs sur l'environnement.

- 2- Une ligne droite horizontale qui représente le temps, sépare les deux couches et référence leurs évolutions.
- 3- L'ensemble du schéma forme une sorte de diagramme de PERT dont le point de synchronisation des tâches est le point central du X.
- 4- Le système forme un tout qui associe une partie matérielle plongée dans l'environnement avec une partie immatérielle, composée de données et du logiciels qui sont mis en œuvre par une partie électronique et informatique embarquée [39].
- 5- L'environnement est représenté en bas du schéma par un rectangle bleu. Il existe avant, pendant et après le système et échange tout au long la vie du système des flux de matières qui sont utilisées pour construire le robot.
- 6- Chaque couche est représentée dans le temps par un cycle de vie qui correspond aux cycles tels que définis par la méthode ACV.
- 7- L'intégration de la partie immatérielle dans la partie matérielle n'y est pas représentée explicitement, mais la méthode considère le produit final concret comme une incorporation du logiciel et des données dans la partie matérielle. Cette incorporation a lieu à la fin des deux phases d'intégration et de production.

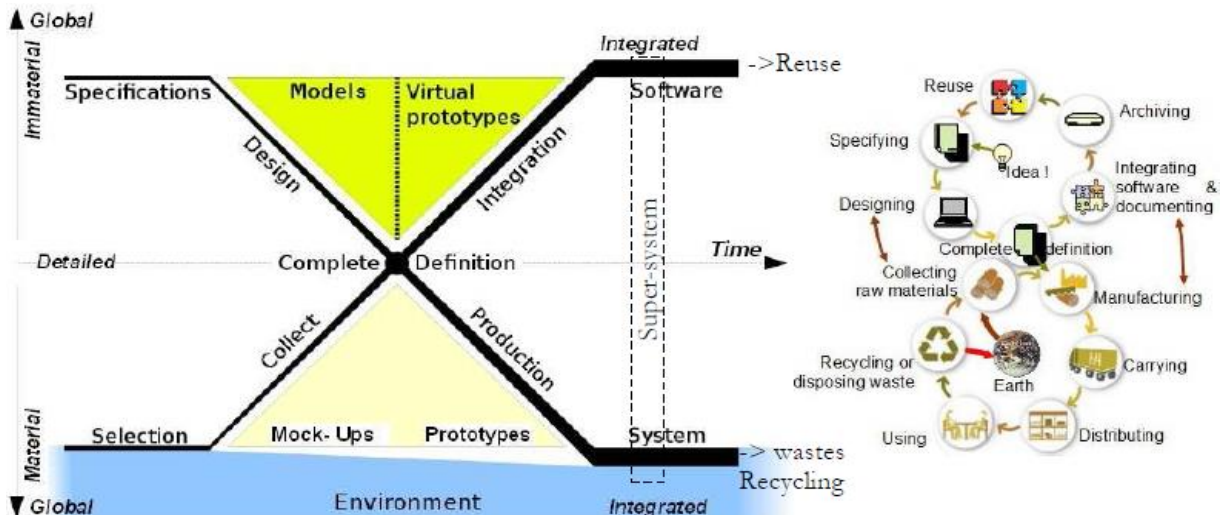


Figure 24 : Le schéma de la méthode en X [38].

4.6. Synthèse du chapitre 4

Nous avons présenté dans ce chapitre les principes de l'ingénierie des modèles qui nous permis de concevoir et développer notre méta modèle SuRo. La majorité des concepts que nous avons intégrés dans SuRo ont été étudiés dans la partie analyse du rapport. Certains autres peuvent encore être enrichis ou adaptés selon le contexte. Cela est possible grâce à la flexibilité qui caractérise le PSM. Ce dernier pourra sans doute, faire preuve de travaux pour être transformé en PSI et produire des simulations de scénarios.

Chapitre 5

Conclusion et Perspective

Dans ce travail, nous avons apporté une première contribution dans le domaine de la robotique durable, pour répondre aux enjeux environnementaux qui vont survenir dans le future et qui vont nous appeler à changer les modèles de conception en robotique. Préserver l'environnement pour les générations futures est une notion fondamentale du développement durable. Ainsi, notre contribution se base sur ce principe, afin de rétablir le lien intrinsèque entre l'être humain et l'environnement, et améliorer leurs relations. Concevoir et développer des robots durables sont des voies scientifiques qui nous permettent d'atteindre ces objectifs.

Dans le premier chapitre, nous avons présenté nos motivations. Elles ont été tirées de faits réels, observés dans la vie courante ; qu'il s'agisse de l'accroissement du marché de la robotique qui est caractérisé par une forte demande ou bien de l'état de l'environnement et ses ressources. Nos hypothèses nous ont toujours suivi du début jusqu'à la fin de ce stage. Elles nous canalisent dans l'élaboration de la ligne de conduite pour atteindre nos objectifs.

Nos investigations concernant les travaux académiques dans le domaine de la robotique durable avec l'enquête auprès des industriels Français en robotique, nous ont permis de constater son état actuel. La robotique durable fait ses premiers pas. Elle bénéficiera sûrement, des expériences acquises en robotique et en développement durable pour les intégrer au mieux, et donc réduire la complexité du domaine.

L'analyse du chapitre 3 a été basée sur l'idée que le robot est composé d'une partie logicielle incorporée dans une partie matérielle. Nous avons présenté tous les éléments essentiels de ces deux parties qui constituent le robot. Le choix des composants logiciels a été motivé par la tendance actuelle en développement de logicielle de robot et des avantages offerts par ce paradigme, tels que la réutilisation, l'ouverture, l'évolutivité et la modularité. Des avantages qui s'inscrivent dans une perspective de développement durable. Nous avons introduit pour chacune des parties ce concept afin de concevoir et développer des robots durables.

Le modèle SuRo est basé sur l'analyse du chapitre 3 et les hypothèses de notre travail. Comme nous l'avons vu, nous avons utilisé les principes d'ingénierie des modèles pour développer notre méta modèle. SuRo considère la notion d'entité comme principe central dans le système, qui est composé d'un environnement avec lequel en fait des échanges principalement. L'entité produit des comportements en se basant sur sa structure qui est composé d'une partie informationnelle et matérielle. L'utilité de SuRo est qu'il permet de faire des simulations de scénarios lorsqu'il sera enrichi. Imaginons un projet de l'agence européenne de l'espace, qui consiste à envoyer un essaim de robots dans la lune. Nous pouvons admettre plusieurs cas :

- Les robots n'ont pas été construits, les parties prenantes cherchent à évaluer leurs choix de conception selon plusieurs contraintes. SuRo permettra de faire des simulations pour connaître les matériaux des robots qui ont moins d'impacts sur l'environnement dans lequel l'essaim va effectuer sa mission, ou bien, il permettra d'exécuter des scénarios de consommation énergétique de la partie logicielle en indiquant les composants logiciels qui consomment plus d'énergie. On peut aussi envisager l'estimation des coûts des flux tout au long le cycle de vie du projet, évaluer ceux des partie prenantes par exemple, pour les réduire.
- Les robots existent déjà, mais on aimerait les augmenter par des composants matériels ou bien, par de nouvelles applications logicielles, en remplaçant leurs anciennes parties. Dans ce cas

SuRo pourra fournir des simulations pour chaque situation et estimer les impacts et les coûts associés. Selon ces estimations les parties prenantes pourront prendre des décisions, pour choisir les meilleures options possibles.

- Si l'un des robots se perd dans son environnement, SuRo pourra évaluer les conséquences de ce scénario, en termes d'impacts et de coûts.

SuRo offrira un outil d'aide à la décision, de prédictions d'impacts et d'estimation de coûts pour améliorer la conception et le développement des robots. Quelles que soient les domaines dans lesquels le robot intervient : agriculture, médecine, exploration, surveillance, construction mécanique, robotique d'assistance, robotique de service, etc.

SuRo à l'état actuel est un noyau, nous envisageons plusieurs perspectives pour l'améliorer : plusieurs concepts peuvent être enrichis en collaborant avec des spécialistes du domaine ; le concept *Environment* peut être augmenté en lui ajoutant des notions d'écologie qui définissent les liens entre l'environnement et les entités. Le concept *Cost* est une porte ouverte qui peut mener vers des analyses financières des coûts des entités et de leurs impacts. En plus, il est possible de profiter de la flexibilité de la solution pour instancier SuRo dans un profile adapté à cette analyse. Le concept *Knowledge* pourra aussi faire sujet de recherche pour lui inclure des notions d'ingénierie des connaissances. Ce point peut être indispensable, dans le cas où on veut que le robot est une abstraction interne de son monde. Dans cette situation, une partie de SuRo lui sera incorporé et lui permettra aussi d'apprendre des situations nouvelles. Le robot pourra adapter ses comportements pour minimiser les impacts sur l'environnement, pour consommer moins d'énergie, etc. Le concept *Phase* pourra être enrichi par les notions de gestion de projet, pour permettre aux parties prenantes de contrôler leurs programmes et leurs tâches et minimiser les coûts. Le concept *StakeHolder* pourra être enrichi pour définir toutes les spécialisations et les catégories possibles d'acteurs et d'organisations qui interviennent dans le cycle de vie du robot, et surtout leurs relations et leurs interactions avec le robot et l'environnement. Dans le concept *SoftwareComponent*, les relations entre ses notions doivent être spécifiées pour permettre de faire des évaluations plus précises. Le concept qualité pourra être spécialisé en lui intégrant les catégories de qualité et leurs métriques (ex : fiabilité : temps moyen entre les pannes, efficacité : consommation énergétique, etc.). Des contrainte OCL doivent aussi être introduites dans SuRo, pour améliorer la cohérence du méta modèle, par exemple un *Stakeholder* est une entité qui n'est pas composée d'un logiciel (à moins qu'il soit un cyborg). Dans l'état actuel tous les *Stakeholders* sont composés d'une partie informationnelle qui peut être spécialisée en un logiciel. Les contraintes OCL doivent permettre d'éviter ce genre d'incohérences. Finalement la transformation du PSM en un code devra être suivie d'un effort minimal de programmation pour pouvoir développer un outil de simulation ou adapter le code selon un contexte particulier.

En parallèle, nous avons aussi commencé à travailler sur un cas concret d'un robot compagnon pour pouvoir le tester SuRo, selon plusieurs scénarios. Les détails des avancées de cette analyse sont dans l'annexe.

Références

- [1] R. Potter, «How to Compete with Offshore Low Labor Costs: Employ Highly Skilled Labor at 30 Cents per Hour», Robots 2004 Conference, Ypsilanti, Michigan, USA, 2004.
- [2] D. Espes, Y. Autret, J. Vareille et P. Le Parc, «Helping and monitoring robots at low cost», 21ème Congrès Français de Mécanique, Bordeaux, France, 2013.
- [3] S. Sun, «How Tesla's 1898 Patent Changed the World», Teleautomaton, 22 Octobre 2010. [En ligne]. <http://teleautomaton.com/post/1373803033/how-teslas-1898-patent-changed-the-world> [Accès le 16 Mai 2014].
- [4] K. Amanda, «Elektro: 1939 smoking robot saved from oblivion», CBS Interactive Inc, 5 Avril 2012. [En ligne], <http://www.cnet.com/news/elektro-1939-smoking-robot-saved-from-oblivion/> [Accès le 16 Mai 2014].
- [5] F. ABB, «ABB Robotics ... 30 ans déjà !», ABB, 2014. [En ligne] <http://www.abb.fr/product/ap/seitp327/a51a8a189e666f35c12573d30056fc9f.aspx> [Accès le 16 Mai 2014].
- [6] IFR, «Industrial Robot Statistics», International Federation of Robotics, 2013. [En ligne] <http://www.ifr.org/industrial-robots/statistics/> [Accès le 19 Mai 2014].
- [7] A. Middendorf, S. Deyter, J. Gausemeier, N. F. Nissen et H. Reichl, «Integration of Reliability and Environmental Aspects in Early Design Stages of Mechatronics», Sustainable Systems and Technology, Phoenix, AZ USA, 2009.
- [8] M. Jouaneh, Fundamentals of Mechatronics, Stamford, USA: Global Engineering: Christopher M. Shortt, 2013.
- [9] G. Branwyn, Absolute Beginner's Guide to Building Robots, USA: Que Publishing, 2003.
- [10] M. Xie, Fundamentals Of Robotics Linking Perception To Action, Singapore: World Scientific Publishing, 2003.
- [11] S. Gibilisco, Concise Encyclopedia of Robotics, New York USA: McGraw-Hill, 2003.
- [12] R. H. Bishop, The Mechatronics Handbook, Austin Texas USA: CRC Press LLC, 2003.
- [13] ADEME, «Management environnemental et éco-produits» [En ligne] <http://www2.ademe.fr/> [Accès le 18 Janvier 2014].
- [14] M. Pradel, «Les analyses du cycle de vie dans le domaine agricole», Irstea-DPV, 2013. [En ligne] <http://www.set-revue.fr/>. [Accès le 18 Janvier 2014].
- [15] D. Riopel, M. Chouinard, S. Marcotte et D. Aitkadi, «Ecoconception», chez Ingénierie et gestion de la logistique inverse: Vers des réseaux durables, Paris, France, Lavoisier, 2011, pp. 139-140-142.
- [16] Université de Valenciennes, «Méthodologie de l'analyse du cycle de vie (ACV)», Juin 2012. [En ligne]. <http://stockage.univ-valenciennes.fr/MenetACVBAT20120704/acvbat/chap03> [Accès le 20 Mai 2014].
- [17] Systemes-Durables, «Analyse du Cycle de Vie», [En ligne]. <http://www.systemes-durables.com/> [Accès le 20 Mai 2014].
- [18] O. Jolliet, M. Saadé, P. Crettaz et S. Shaked, Analyse du cycle de vie : Comprendre et réaliser un écobilan, Paris: PPUR, 2010.
- [19] F. Ingrand, «Architectures Logicielles pour la Robotique Autonome», Journées Nationales de Recherche en Robotique, 8-10, Octobre 2003.
- [20] R. Passama, «Conception et développement de contrôleurs de robots : Une méthodologie basée sur les composants logiciels», Université de Montpellier II, Montpellier, France, 2006.
- [21] R. A. Brooks, «A robust layered control system for a mobile robot», Artificial intelligence at MIT : expanding frontiers, pp. 2-27, 1990.
- [22] A. El Jallaoui, «Gestion contextuelle de tâches pour le contrôle d'un véhicule sous-marin autonome», Université De Montpellier II, Montpellier, 2007.
- [23] D. Filliat, «Robotique Mobile», ENSTA ParisTech, Paris France, 2011.
- [24] M. Völter, «A Taxonomy of Components», Journal of Object Technology, vol. 2, n° 14, pp. 119-125, 2003.

- [25] M. Dick et S. Naumann, «Enhancing Software Engineering Processes towards Sustainable Software Product Design», *EnviroInfo 2010: Integration of Environmental Information in Europe*, Cologne/Bonn, Allemagne, 2010.
- [26] S. Naumann, E. Kern et M. Dick, «Classifying Green Software Engineering - The GREENSOFT Model», Allemagne, 2013.
- [27] M. Dick, J. Drangmeister, E. Kern et S. Naumann, «Green Software Engineering with Agile Methods», *GREENS 2013*, San Francisco, CA, USA, 2013.
- [28] F. Berkhout et J. Hertin, «Impacts of Information and Communication Technologies on Environmental Sustainability: speculations and evidence», Université du Sussex, Brighton, RU, 2001.
- [29] P. Bozzelli, Q. Gu et P. Lago, «A systematic literature review on green software metrics», Amsterdam, Pays bas, 2013.
- [30] B. Combemale, «Ingénierie Dirigée par les Modèles (IDM): état de l'art», Institut de Recherche en Informatique de Toulouse (UMR CNRS 5505), Toulouse, France, 2008.
- [31] J. Bézin et X. Blanc, «MDA : vers un important changement de paradigme en génie logiciel», <http://www.devreference.net/>, France, 2002.
- [32] J. Bézin, «In Search of a Basic Principle for Model Driven Engineering», *UPGRADE: European Journal for the Informatics Professionals*, vol. V, n° 12, pp. 21-24, 2004.
- [33] S. Andre, «MDA (Model Driven Architecture) principes et états de l'art», Conservatoire National Des Arts Et Métiers, Lyon, France, 2004.
- [34] C. Schlegel, A. Steck et A. Lotz, «Model-Driven Software Development in Robotics: Communication Patterns as Key for a Robotics Component Model», *Introduction to Modern Robotics*, Ulm, Allemagne, iConcept Press, 2012, pp. 119-150.
- [35] Eclipse, «Eclipse Modeling Framework Project (EMF) », The Eclipse Foundation, 2014. [En ligne] Available: <http://www.eclipse.org/modeling/emf/> [Accès le 29 Mai 2014].
- [36] A. Touil, «Vers un langage de modélisation spécifique au domaine des systèmes de télécontrôle ubiquitaire», Université de Bretagne Occidentale, Brest, France, 2011.
- [37] E. Merks et J. Sugrue, «EssentialEMF,» DZone, [En ligne] <http://refcardz.dzone.com/refcardz/essential-emf> [Accès le 03 Juin 2014].
- [38] J. Vareille, M. Tahan et C. Languenou, «Une méthodologie de conception basée sur l'environnement : la méthode de développement en X», 21ème Congrès Français de Mécanique, Bordeaux, 2013.
- [39] M.Tahan, A.Touil, J.Vareille et P.Le Parc, «La méthode de développement en X, un autre point de vue sur le cycle de vie», Brest, France, 2012.
- [40] N. Rousse, «IDM Ingénierie Dirigée par les Modèles», Modelia, Toulouse, France, 2007.
- [41] F. Y. VILLEMIN, «Méthodologie Avancée d'Informatisation : L'architecture dirigée par les modèles», CNAM, Paris, France, 2013.
- [42] C. Schlegel, A. Steck et A. Lotz, «Robotic Software Systems: From Code-Driven to Model-Driven Software Development», *Robotic Systems : Applications, Control and Programming*, Ulm, Allemagne, InTech, 2012, pp. 473-502.
- [43] E. Foundation, «Package org.eclipse.emf.ecore» [En ligne] <http://download.eclipse.org/modeling/emf/emf/javadoc/2.9.0/org/eclipse/emf/ecore/package-summary.html> [Accès le 3 Juin 2014].
- [44] ETSI, 'Environmental Engineering (EE); Life Cycle Assessment (LCA) of ICT equipment, networks and services, General methodology and common requirements', ETSI TS 103 199 V1.1.1, ETSI, 2011.
- [45] Orgelet J., 'Influence of the methodological and market changes on the update of LCI dataset - Difficulties and impacts - The example of EEE products', Life Cycle in Practice conference, Lille, 2013.
- [46] Orgelet J., Prunel D., Durieux X., 'Applicability, contribution and limitation of new international standards for IT services environmental footprint' SETAC LCA case study symposium, Copenhagen, 2012.
- [47] Tchakaloff, B., Saudrais S., Babau J.P., 'ORQA: Modeling Energy and Quality of Service within AUTOSAR Models', *Quality of Software Architectures (QoSA)*, Vancouver, 2013.

Annexe 1

Formulaire de l'enquête sur la robotique durable

Enquête sur la robotique durable

Cette enquête a pour objectif de connaître les pratiques environnementales des compagnies françaises de robotique pour servir dans mon travail de stage sur la conception et le développement de robots durables qui a pour objectif de créer un outil qui permettrait d'améliorer les performances environnementales des robots. Nous vous remercions pour votre collaboration.

Informations générales

Q1. Quel est le nom de votre entreprise?

Q2. Dans quels domaines interviennent les robots que vous construisez?

Q3. Avez-vous une stratégie environnementale intégrée dans votre compagnie?

Oui Non

Q4. Quelles directives et normes environnementales sont prises en considération?

- ROHS
- DEEE
- ISO 140XX
- EPEAT
- Autres

Développement durable

Q5. Utilisez-vous les principes d'écoconception pour la conception des robots?

Oui Non

Q6. Quels aspects d'écoconception sont pris en considération?

- Réduction de la consommation d'énergie
- Réduction de la consommation de matières premières
- Réduction des émissions dans l'environnement
- Optimisation de la durabilité
- Autres

Q7. Quelles propriétés environnementales ont les produits composants vos robots?

- Recyclés
- Recyclables
- Naturels
- Biodégradables
- Réutilisables
- Renouvelables
- Peu consommateurs d'énergie
- Autres

Q8. Estimez-vous les impacts environnementaux des robots?

- Oui Non

Q9. A quel niveau le faites vous?

- Matériel
- Logiciel
- Matériel et Logiciel

Q10. Utilisez-vous l'ACV (L'analyse de cycle de vie)?

- Oui Non

Q11. Prenez-vous en compte l'ACV du logiciel?

- Oui Non

Q12. Si vous le faite, quelles sont les contraintes de la couche logicielle prises en considération?

An empty text input field with a light gray background and a thin border. It features standard scrollbars on the right and bottom edges.

Q13. Quels indicateurs de performances environnementales utilisez-vous?

- Indicateurs de l'ISO 14040
- Ecoindicator
- Recipe
- Autres

Q14. Quels outils logiciels utilisez vous pour améliorer les performances environnementales des robots?

An empty text input field with a light gray background and a thin border. It features standard scrollbars on the right and bottom edges.

Q15. Vos robots sont modulaires au niveau:

- Matériel
- Logiciel
- Matériel et Logiciel
- Non

Q16. Vos robots sont maintenables au niveau:

- Matériel
- Logiciel
- Matériel et Logiciel
- Non

Q17. A quels niveaux Intégrez-vous la fiabilité?

- Matériel
- Logiciel
- Matériel et Logiciel
- Non

Q18. Publiez-vous les données de fiabilité?

Oui Non

Q19. Ou se trouvent ces publications?



Annexe 2

ACV d'un robot compagnon

Analyse du cycle de vie d'un robot compagnon

Le contexte d'utilisation du robot est montré sur la figure 1. Les composants sont présentés sur la figure 2, ils sont choisis sur étagère, à l'exception du capot supérieur. Le robot est constitué principalement d'une plate-forme mobile Rover 5, qui comprend un châssis, deux chenilles, des moteurs, des capteurs, des roues et leurs essieux et des connecteurs. Connectée à cette plate-forme nous avons une unité centrale composée d'une carte Arduino, d'un module Bluetooth et d'une carte de relais. Pour alimenter l'ensemble nous utilisons deux accus Li-Polymère. Sur le capot, nous plaçons un smartphone sous Android.

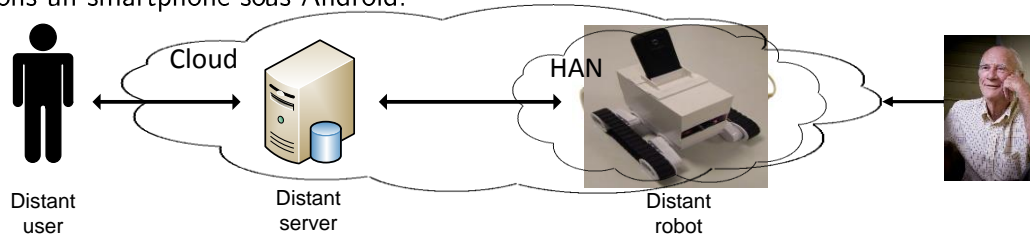


Figure 1 : le contexte d'utilisation, les composants du robot.

La seconde configuration de l'unité centrale (non représentée) est composée par un Raspberry Pi, un module WiFi-USB et un module USB-Phidget. Ces deux configurations sont suffisantes pour développer le logiciel pour les services attendus. Dans un précédent article [2], notre équipe a expliqué la méthodologie utilisée pour concevoir le robot, choisir les éléments à l'égard des critères de performance en partant de l'unité fonctionnelle. Cette dernière est définie comme suit : par période d'une année d'utilisation, une distance franchissable de 1500 km/an, un contrôle distant permanent, une masse maximale de environ 3 kg, une taille au sol équivalente à deux chaussures, un MTTRi idéal de quelques minutes, un MTBFm minimum de 1 an, la capacité de communiquer images et sons par un réseau sans fil avec une connexion de largeur de bande supérieure à 1Mbits/s, etc.



Figure 2 : les composants du robot.

Les limites du système comprennent la plate-forme mobile, l'unité centrale, les piles, et le logiciel. Nous considérons comme au-delà des limites le nuage local (Home Area Network – HAN) et le smartphone, bien que les caméras et les capteurs soient pour la plupart inclus dedans. Pour nous, il constitue la charge utile. Nous pensons que dans l'avenir les véhicules seront entièrement automatiques, alors la directive européenne VHU sera appliquée pour les deux : véhicules et robots mobiles, ce qui impliquera un taux de recyclage et réutilisation des éléments supérieur ou égal à 95%.

Pour effectuer l'ACV, nous avons recueilli des données sur chaque composant : masse, les emplacements du producteur, du distributeur, la consommation d'énergie, le respect des directives européennes. Mais certaines données sont manquantes comme la durée de vie, la possibilité de réutilisation, les compositions détaillées des différentes sous-parties, etc. Nous avons essayé d'utiliser les données du catalogue "TraceParts", qui contient la bibliothèque accessible la plus grande de composants mécatroniques et l'application "Bilan Produit" (ADEME).

Les consommations d'énergie sont communiquées par les fournisseurs ou les forums d'utilisateurs. Celle de la solution Arduino en utilisation est plus faible que celle de l'autre, toutefois les deux sont inférieures à la puissance requise par la plate-forme mobile.

Les données manquantes ne permettent pas d'effectuer une analyse de cycle de vie précise. Mais tous les composants électriques et électroniques sont fabriqués en utilisant des technologies similaires, les valeurs manquantes pourraient être extrapolées à partir des résultats de travaux référencés [7] [44] [45], en les pondérant par des rapports de masse. Donc, il apparaît possible de faire une évaluation du cycle de vie attributionnelle (A-LCA) d'un robot dans les deux configurations. Nous constatons que l'unité centrale sur la base de l'Arduino avec ses deux cartes ajoutées, a une masse moitié de celle de la solution RPi (de 197g), cette dernière ne représentant que le sixième de la masse totale (1,250 kg - de 1,148 kg). Les autres impacts pourraient être évalués de la même manière.

Dans le cas du logiciel [46], le fait important est qu'il peut être modifié pendant la durée de vie du robot. Certains travaux montrent que le comportement du robot peut être modifié en cours d'utilisation pour réduire dynamiquement sa consommation d'énergie [47]. Nous proposons d'effectuer une sorte d'analyse du cycle de vie conséquentielle (C-LCA). Par exemple, nous pouvons procéder par comparaison entre des versions différentes du logiciel offrant les mêmes services, par des mesures de consommation et de durée de vie des composants et des matériaux, ainsi que l'impact du développement et la distribution de ces versions de logiciels.

Composant	masse	Matériaux principaux	Puissance électrique consommée	Recyclabilité	Origine - revendeur	Durée de vie	Emission ondes EM
Rover 5	865g	ABS, Cu, NdFeB, époxy, PVC	~10W selon le contexte	95% (VHU)	Dagu – Chine Gotronic Fr	nc	nc
Accus Li-Ion-Polymer	2x94g	LiCoO2 LiC	2200 mAh 7,4V	nc	Chine	nc	nc
Arduino	28g	Epoxy, Cu, Si	0,1W à 5W	nc	It	nc	nc
BlueTooth	17g	Epoxy, Cu, Si	Par l'Arduino	nc	It	nc	2,45 GHz
Relais	50g	Epoxy, Cu, Si	Par l'Arduino	nc	It	nc	nc
RaspBerry Pi	42g	Epoxy, Cu, Si	3,5W à 6W	nc	Uk	nc	nc
USB Wifi	55g	Epoxy, Cu, Si	Par le Raspberry Pi	nc	Uk	nc	2,45 GHz
USB Phidget	100g	Epoxy, Cu, Si	Par le Raspberry Pi	nc	Uk	nc	nc

Tableau 1 : les composants du robot compagnon et leurs caractéristiques connues

nc = non communiqué