

Réalisation d'outils de qualification des
applications et des services robotiques

Margaux CHARLES

P14

Table des matières

Chapitre 1

Remerciements

Je remercie l'UTC de m'avoir permis de réaliser ce stage au cours de ma formation et M. Alessandro VICTORINO pour son attention en tant que suiveur pédagogique.

Je tiens à remercier la société Aldebaran et tout particulièrement Charles POISSON pour leur confiance en me proposant un stage aussi intéressant que valorisant.

Merci également à William ROUIE pour m'avoir transmis sa rigueur et ses connaissances avec patience et gentillesse.

Merci à Ratiba Kitty ROZE pour ses précieux conseils et sa bonne humeur au quotidien.

Merci à Thomas CHARLES de m'avoir fait découvrir cette entreprise.

Je remercie l'ensemble de la Qualification de m'avoir considéré comme un membre à part entière de leur équipe. Ils m'ont permis d'apprendre de nombreuses choses en me proposant de réaliser des projets variés.

Et enfin, merci à tous les Aldebariens qui m'ont fait découvrir leur passion de la robotique et m'ont accueilli chaleureusement durant ces 6 mois.

Chapitre 2

Abréviations

ADE : Application Distribution Engine
API : Application Programming Interface
CPU : Central Process Unit
CSV : Comma-Separated Values
DCM : Device Communication Manager
GPL : General Public License
GUI : Graphic User Interface
HAL : Hardware Abstraction Layer
HTML : HyperText Markup Language
IDE : Integrated Development Environment
IP : Internet Protocol
LED : Light-Emitting Diode
OS : Operating System
PEP : Python Enhancement Proposal
REST : Representational State Transfer
RHM : Robot Health Monitor
RPS : Robotic Preference System
SAV : Service Après-Vente
SDK : Software Development Kit
SFTP : SSH File Transfer Protocol
SSH : Secure Shell
TCP : Transmission Control Protocol

UCS : Universal Character Set

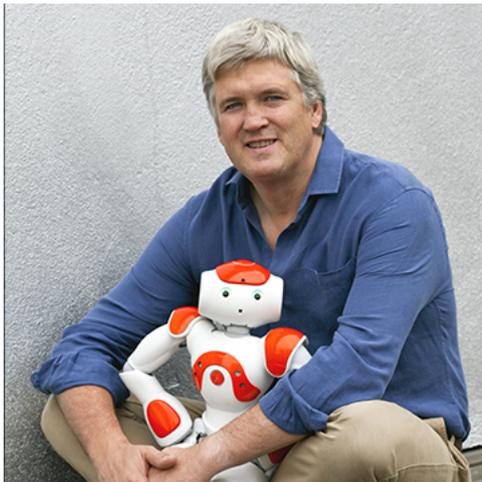
UML : Unified Modeling Language

UTF-8 : UCS Transformation Format - 8 bits

Chapitre 3

Présentation

3.1 L'entreprise



Aldebaran Robotics est aujourd'hui le leader mondial de la robotique humanoïde. Cette start-up créée en 2005 par Bruno Maisonnier a su développer la magie des robots. Elle a pour vocation de les rendre interactifs, évolutifs et abordables afin d'aider les humains dans leur quotidien. Les produits développés sont à la fois beaux, et simple d'utilisation.

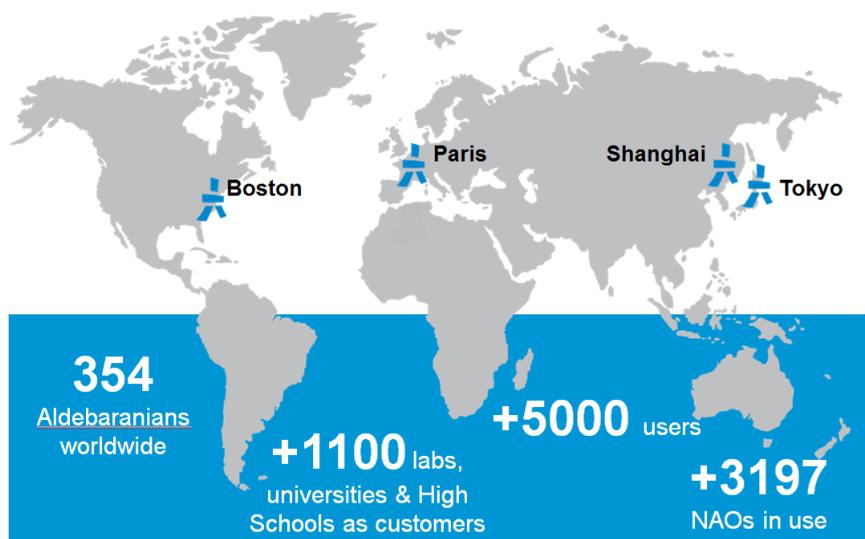
3.2 Pourquoi Aldebaran Robotics ?

Aldebaran est le nom de l'étoile la plus brillante de la constellation du Taureau. Elle était utilisée autrefois pour la navigation et l'orientation. Elle est donc la métaphore de la voie à suivre, d'un nouveau territoire à découvrir : le monde de la robotique.

3.3 Philosophie de l'entreprise

Développer la magie des robots et l'entretenir est l'objectif d'Aldebaran Robotics. Le succès de l'entreprise est due à la grande implication de ses employés qui sont pour la plupart des passionnés. Ils sont encouragés à faire preuve d'initiative, de pragmatisme et d'imagination.

3.4 Aldebaran dans le monde



Aldebaran Robotics est aujourd'hui présente dans 4 régions du monde : Paris, Boston, Shanghai, et Tokyo. Cependant, elle reste une start-up française, et la majorité de son effectif travaille à Paris.

3.5 Actualités

SoftBank est le premier opérateur mobile au Japon. Le groupe partage la même conviction que Aldebaran Robotics : la robotique personnelle est la prochaine révolution technologique comme a pu l'être le PC ou internet dans le passé . Aldebaran Robotics a donc été choisi par SoftBank pour développer un robot capable d'accueillir et de divertir les visiteurs des magasins du groupe : Pepper.

Le 5 Juin 2014, Pepper a été présenté lors d'une conférence de presse à Tokyo. Aldebaran Robotics a annoncé qu'il rejoignait le groupe SoftBank et a changé de nom pour Aldebaran.



Depuis deux ans SoftBank est devenu l'actionnaire majoritaire d'Aldebaran Robotics. Cet investissement a permis à Aldebaran d'atteindre plus rapidement son objectif : commercialiser des robots compagnons humanoïdes pour le grand public, à un prix abordable.

3.6 La famille Aldebaran

3.6.1 NAO



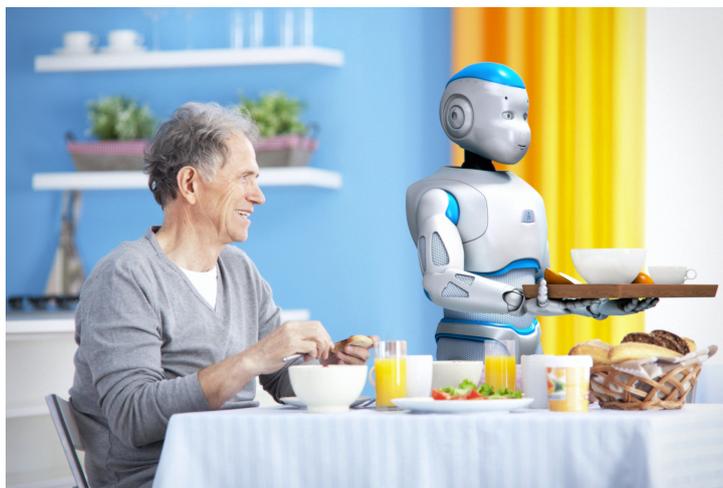
NAO est un robot humanoïde de 58 cm de haut. Il est né en 2006 et il est commercialisé depuis 2008. Il a été vendu à plus de 5000 exemplaires dans 70 pays du monde.

Il est destiné à devenir un compagnon, un membre de la famille ou encore un assistant aux personnes âgées. Il est également utilisé dans le milieu académique pour apprendre aux étudiants la programmation de robots. Un logiciel destiné aux enfants autistes est disponible pour NAO.

L'interaction entre le robot et l'utilisateur est très simple : il suffit de lui parler, il va vous comprendre.

Ce robot est capable d'analyse et d'apprentissage. Il est proactif : c'est à dire qu'il est capable de dispenser des services sans qu'on lui demande, simplement en analysant son environnement, vos émotions ou même vos habitudes de vie.

3.6.2 Roméo



2007 : Bruno Maisonnier a eu l'idée de développer un robot humanoïde de grande taille pouvant ainsi être à la fois un compagnon et un véritable assistant à la personne.

2009 : Le projet Roméo commence. Il s'agit d'un projet collaboratif entre des laboratoires et des industriels comme Aldebaran.

Mars 2014 : Roméo a été présenté pour la première fois au grand public au salon de Lyon Innorobo.

Roméo est un robot humanoïde destiné à l'aide et l'assistance aux personnes âgées ou en perte d'autonomie. Il est actuellement utilisé par des laboratoires européens comme plateforme de recherche mais il devrait être présent dans les maisons de retraite dès 2017.

Il mesure 1m40, soit la taille d'un enfant de 8 ans et pèse 40kg. Sa structure en fibre de carbone et caoutchouc le rend assez léger pour ne pas blesser la personne qu'il va assister. Ce robot est capable de marcher, et de parler. De plus sa vision en 3D lui permet de reconnaître des objets. Ses yeux sont mobiles contrairement à NAO. Il devrait bientôt pouvoir poser des objets sur une table, ouvrir des portes et monter des escaliers.

3.6.3 Pepper



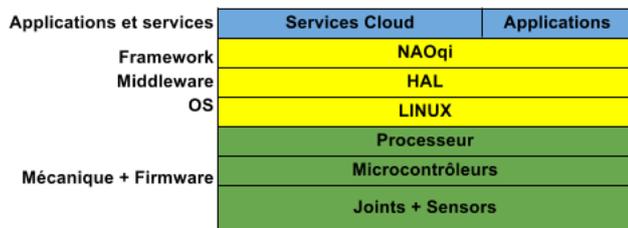
Pepper est le premier robot émotionnel. Il mesure 1m20 et pèse 28kg. Il est capable de comprendre 80% d'une conversation et possède une autonomie de 12h.

Ce robot a été créé dans le but de rendre les gens heureux et d'interagir avec eux. Il est capable d'interpréter les émotions humaines. Il peut détecter 5 expressions sur le visage humain : le sourire, le froncement de sourcil, la surprise, la colère, et la tristesse.

Pepper est aujourd'hui capable de parler 4 langues : français, anglais, espagnol et japonais. Il est visible depuis le 6 juin 2014 dans les magasins SoftBank à Tokyo. Sa commercialisation au Japon est prévue pour 2015 au prix de 1500 euros.



3.7 Lexique

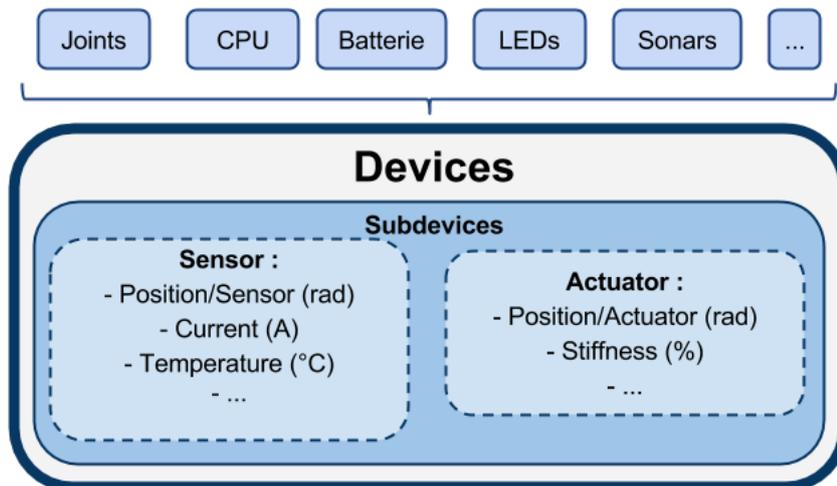


Hardware : Ensemble des éléments physiques employés pour le traitement de l'information par opposition au logiciel.

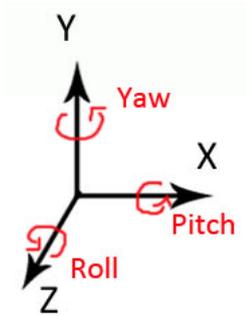
Firmware : C'est un ensemble d'instructions et de structures de données qui sont intégrées dans du matériel informatique pour qu'il puisse fonctionner.

Middleware : Un middleware est un logiciel qui permet à différentes applications d'échanger et d'inter-opérer. Ici il s'agit du HAL, la couche d'abstraction hardware entre les softwares de haut niveau utilisés par la carte mère des robots et les microcontrôleurs (devices).

Software : Logiciel



Joint : Chaque degré de liberté du robot est un joint. La convention de nommage est la suivante : nom de l'articulation + mouvement de rotation selon la convention Roulis, Tangage, Lacet (en anglais : Pitch, Yaw, Roll) utilisée en aéronautique (ex : HeadYaw, HeadPitch...).



Device : C'est un microcontrôleur assurant la communication entre les sub-devices et le HAL. Il est relié à des subdevices qui sont des actionneurs ou des capteurs.

Subdevice : Les subdevices sont des capteurs ou des actionneurs reliés à des devices. Ils possèdent tous un nom et une valeur principale Value qui est la valeur du capteur ou de l'actionneur.

Actuator : Un actionneur est un subdevice capable de recevoir des commandes. La dernière commande qui lui a été envoyée est stockée dans une valeur ALMemory. Par exemple pour un joint comme HeadYaw, sa valeur va être sa position souhaitée (angle en radian).

Sensor : Un sensor est un capteur. Dans ALMemory, la valeur d'un sensor pour un joint va être sa valeur actuelle. Par exemple pour une articulation comme HeadYaw, sa valeur va être sa position réelle (angle en radian).

Les principaux modules :

ALMemory : Ce module contient les valeurs des sensors et des actuators ainsi que les différents événements.

DCM : Ce module actualise les valeurs dans ALMemory et envoie des commandes aux actuators. C'est un module NAOqi qui appartient plus particulièrement à la couche d'abstraction Hardware (HAL). Il assure la communication entre l'ensemble des devices du robot exceptés les entrées et sorties de son et les caméras.

ALMotion : Ce module fournit l'ensemble des fonctions permettant de faire bouger le robot. Il contient les commandes qui contrôlent la souplesse, l'angle des différents joints du robot ainsi que sa marche.

Proxy : Objet qui contient l'ensemble des méthodes du module qu'il représente.

3.8 Les offres

3.8.1 NAOqi

C'est le framework développé par Aldebaran, basé sur l'interaction naturelle et l'émotion. C'est le logiciel qui donne vie au robot. Il contient des modules pour gérer les services de bases du robot comme l'audio, les déplacements, la vision...

3.8.2 Choregraphe

C'est un outil de programmation graphique permettant de programmer et d'animer le robot sans connaissance préalable en programmation. Cet outil permet de rendre plus intuitive, et accessible la conception d'applications pour NAO.

3.8.3 Monitor

C'est un outil qui permet de connaître l'état du robot en affichant la caméra, les valeurs de ALMemory, la température des moteurs, ou encore sa position...

3.8.4 SDK

C'est l'ensemble des outils et des bibliothèques permettant de programmer le robot que l'on soit un programmeur novice ou expérimenté : Chorégraphe, Monitor... Il est multi-plateforme et est compatible avec de nombreux langages et plates-formes robotiques comme Java, Javascript, Matlab, C++, Python, Urbi, .Net...

3.8.5 NAO Webot

Ce logiciel de simulation 3D permet de développer sans avoir un vrai robot. Les applications sont testées dans un environnement virtuel régi par les lois physiques réelles et modifiable par l'utilisateur (intégration d'objets...).

3.9 Les services

3.9.1 STORE

Forum de partage entre les membres de la communauté d'utilisateurs de NAO et de Pepper. Ils peuvent s'y échanger, vendre ou acheter des applications.

3.9.2 DEVELOPERS WEBSITE

Site permettant aux développeurs de télécharger les mises à jour des softwares, mais aussi de créer et partager des programmes avec la communauté.

3.9.3 USER WEBSITE

Site réservé aux académiques, c'est-à-dire à la recherche et l'éducation. Ils ont accès à un blog, des forums techniques, et des ressources comme des logiciels, ou des tutoriels...

3.9.4 CUSTOMER CARE MONITOR

Service d'Aldebaran en charge de résoudre tous les problèmes que les clients rencontrent avec leur robot.

3.10 Le service Qualification



Aldebaran développe des robots mais également un ensemble de services qui leurs sont associés.

Parmi ces services il y a :

- les applications : elles peuvent être de différents types (danses, jeux, dialogues, etc.) ;
- le SDK ;
- le Web and Cloud : ensemble des services internes et externes permettant de gérer des robots.

Le service Qualification veille donc à la qualité de l'ensemble de l'offre : le produit et les services.

Le service Applications est chargé de tester les applications pour Pepper et NAO.

Le service Web and Cloud qualifie les différents services offerts avec les robots et mais également des outils internes.

J'ai effectué mon stage au sein du service Tools and Investigations. Il a pour rôle d'aider les autres services de qualification en développant des outils.

Chapitre 4

Stage

4.1 Objectif

L'objectif de mon stage a été d'aider les équipes de qualification dans leur travail avant le lancement de Pepper. J'ai donc réalisé de nombreux outils répondant à leurs besoins.

4.2 Méthodes

Les outils sont réalisés en Python afin de permettre un développement rapide tout en facilitant la maintenance.

La méthode de développement utilisée est la méthode agile. À partir de l'expression du besoin, l'objectif est de développer au plus vite une première version de l'outil quitte à ne pas en implémenter toutes les fonctionnalités et de la soumettre aux futurs utilisateurs. Cela permet d'obtenir de premiers retours.

Dans la prochaine version, de nouvelles fonctionnalités seront ajoutées et les bugs seront corrigés. Cela permet d'impliquer l'utilisateur final dans le processus de développement et de créer un consensus. Les applications seront utilisées si elles correspondent aux attentes de l'utilisateur. De plus, les fonctionnalités n'étant pas toutes implémentées en même temps, l'utilisateur apprend à les utiliser progressivement.

4.3 Outils de travail

4.3.1 Gestionnaire de versions

 **git** Git est un gestionnaire de version décentralisé et sous licence libre.

4.3.2 IDE

 **NINJA** Ninja est un IDE conçu pour le développement d'applications Python. Il met en évidence les erreurs de syntaxe et PEP 8 (normes de codage recommandées afin d'encourager toute la communauté Python à coder de la même façon).

4.3.3 Terminal

 Terminator est un terminal sous licence libre GPL. Il offre de nombreux avantages comme pouvoir modifier les couleurs, ou permettre d'ouvrir plusieurs terminaux en divisant une fenêtre.

4.3.4 OS

 C'est un système d'exploitation fondé sur la distribution Linux. Il est gratuit et est composé de logiciels libres.

4.3.5 Language de développement

 **python** Python est un langage de programmation objet, multi-paradigmes et multi-plateformes. Il est sous licence libre et est un excellent langage de prototypage.

4.3.6 Gestion de projets

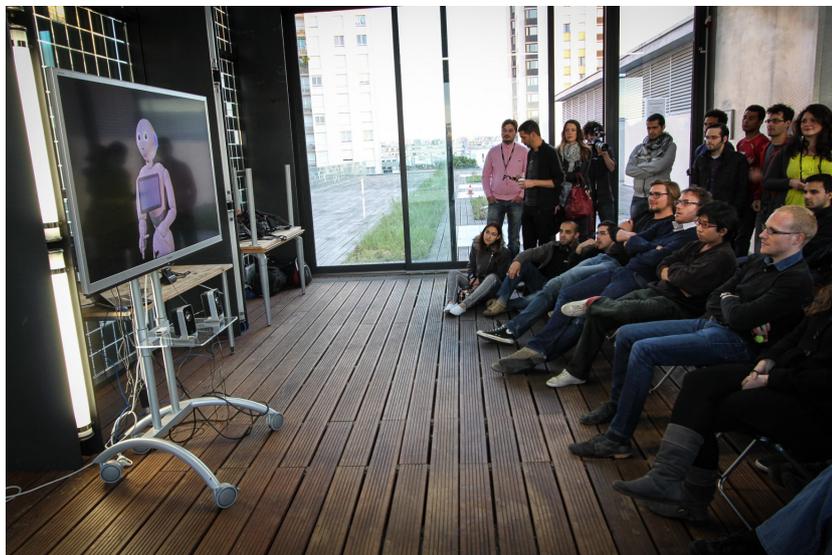
 **REDMINE** Application Web libre de gestion de projets.

Chapitre 5

Conclusion

Au début du stage j'ai eu du mal à rassembler et appliquer toutes les notions étudiées en cours. J'ai donc fait beaucoup de recherches sur internet et j'ai demandé de l'aide et des conseils techniques à mes collègues. J'ai énormément appris et je me suis perfectionnée en Python, langage dont je ne connaissais que les bases.

Mon travail a été très varié, j'ai appris à être plus autonome et à faire preuves d'initiatives.



Durant mon stage j'ai eu la chance d'assister à un moment très important : celui du lancement de Pepper. Toute l'entreprise est venue assister à la retransmission du lancement en direct de Tokyo à 6h le jeudi 5 juin. La confidentialité à laquelle les employés étaient soumis depuis près d'un an a été

levée, ils ont donc enfin pu parler de Pepper à leurs proches. Cette journée a renforcé le sentiment d'appartenance à l'entreprise, et de fierté d'être un Aldebarien. Les différents services ont travaillé très dur les semaines précédant le lancement et tous ont contribué à mener à bien le projet Pepper.



Ce premier stage assistant ingénieur m'a permis d'utiliser toutes mes connaissances acquises depuis le début de ma formation d'ingénieur en informatique à l'UTC. Il a été pour moi très intéressant de passer de la théorie à la pratique. J'ai appris énormément de nouvelles techniques. J'ai aujourd'hui plus de recul sur la théorie vue en cours, ce qui me donne une vision globale de l'ensemble de mes connaissances.

J'ai également énormément appris sur l'organisation de la vie en entreprise, et des services. J'ai apprécié d'être considérée comme un membre à part entière de l'équipe Qualification et j'ai eu la chance de découvrir le monde de la robotique entourée de passionnés.

L'IHM et l'IA sont des domaines qui m'intéressent beaucoup et ce sont les aspects interactifs et proactifs des robots qui ont premièrement retenus mon attention. Cependant, au fur et à mesure de mon stage, j'ai compris également leur fonctionnement d'un point de vue hardware.