

Cahier des charges

Lorthioir Guillaume, Billod Nicolas

Mars 2017

Contexte :

La robotique et l'intelligence artificielle étant deux domaines en plein essor, il est naturel pour les laboratoires de recherche en informatique de se doter des outils adéquats pour contribuer à l'avancée des ces deux domaines.

Dans ce contexte, le Laboratoire d'Informatique de Paris 6 (LIP6) a fait l'acquisition récente de robot Thymio 2. Il lui faut donc développer une architecture logicielle pouvant être supportée par ces robots et permettant de les contrôler. Cette architecture de contrôle devra permettre d'implémenter des comportements plus au moins complexes au sein des Thymios et être assez souple pour pouvoir s'adapter à d'autre types de robots.

L'espace mémoire fourni par les Thymios étant insuffisant, l'implémentation de l'architecture de contrôle se fera sur des nano-ordinateurs Raspberry Pi 3.

Besoins du client :

Objectif

L'objectif principal de ce projet consiste à développer une architecture de contrôle délibérative qui puisse permettre à des robots Thymio II couplés à des Raspberry Pi 3 de remplir des missions coopératives dans un environnement incertain et dynamique.

Architecture

Nous souhaitons que les robots agissent de façon autonome dans un environnement dynamique en utilisant une architecture de contrôle adaptée. Celle-ci comprendra au moins un module de perception et un module de délibération, exploitants les connaissances de l'environnement dans le but de déterminer les commandes à appliquer sur les effecteurs.

Nous souhaitons que l'architecture de contrôle implémentée soit une architecture "à couches", allant du plus bas-niveau (où les fonctions ne font que récupérer les informations des capteurs) au plus haut-niveau (où les fonctions réalisent des algorithmes plus complexes, comme la cartographie de l'environnement). Les fonctions de chaque niveau seront répertoriées dans un document annexe.

Perception

La perception de l'environnement par le robot s'effectuera à l'aide des capteurs infrarouge dont il est pourvu. Les valeurs réceptionnées par ces capteurs seront récupérées par l'ordinateur à travers le D-Bus et seront ensuite exploitées de manière à fournir une représentation du monde adéquate.

Déplacements

Les premières capacités de délibération mise en place seront des fonctions d'évitement d'obstacles et de navigations vers une destination donnée. L'incertitude liée aux déplacements peut être un problème, et ne doit pas être ignorée. Suite à cela, il pourra être envisagé de mettre en place des algorithmes déjà existants pour cartographier l'environnement au fur et à mesure des déplacements du robot.

Coopération

Les robots doivent aussi être en mesure de communiquer et de se coordonner pour réaliser des tâches coopératives, telles que l'exploration, la patrouille, etc. L'implémentation de ces fonctions ne fait pas partie de ce projet mais l'architecture mise en place devra prévoir l'intégration des informations reçues par message et l'exploitation, dans le module délibératif, de celles-ci.

Contraintes :

Le développement de ce projet est soumis à plusieurs contraintes :

- L'architecture de contrôle doit être programmé en langage C.
- Le code développé doit être open-source.
- Celui-ci sera disponible sur le compte gitLab du projet ThymSMA.
- La documentation technique et fonctionnelle du code fournie devra être écrite en anglais.
- L'architecture de contrôle développée est destinée à des robots Thymios 2, mais devra être compatible avec d'autre type de robots.
- Celle-ci devra être supportée par des nano-ordinateurs de type Raspberry Pi 3.
- La communication entre le robot Thymio et le Raspberry Pi 3 s'effectuera à l'aide du D-Bus et de l'interface asebamédulla.
- Un module de communication pour interagir avec un autre projet en développement est à prévoir.
- Des modules complémentaires doivent pouvoir être ajoutés sans perturber l'architecture initiale.

Solution envisagée :

Nous envisageons d'implémenter une architecture "à couches" en nous inspirant des écrits de Rodney A. Brooks, c'est-à-dire que les différentes couches de l'architecture suivront une certaine hiérarchie : si le robots souhaite exécuter un comportement complexe, la couche de plus haut niveau sera en charge d'appeler cette fonction qui, elle-même, appellera des fonctions appartenant à une couche de plus bas niveau, qui à leur tour appelleront des fonctions d'un niveau encore plus bas et ainsi de suite jusqu'à atteindre les fonctions de bases déjà implémentées dans le robot Thymio. Nous développerons cette architecture en C, et les différents niveaux de celle-ci seront répartis sur différents fichiers, c'est la manière dont les fichiers se référenceront les uns par rapport aux autres qui définira l'architecture en question.

La dernière transition (entre les fonctions du robot et les fonctions de notre code) se fera très probablement avec l'aide de la librairie fournie par GDBus.

La création d'un support amovible se fixant au robot Thymio nous permettra d'y installer le Raspberry Pi ainsi que son alimentation pour pouvoir permettre l'autonomie du robot. Ce support permettra aussi d'insérer d'autres périphériques, comme par exemple une caméra pour permettre une meilleur identification de l'environnement.