# **Voice IT :** approche sémantique du controle vocal d'objets connectés

Yassin Chabeb,
**Palo-it**

*Yassin CHABEB, consultant R&D et IT Yassin justifie de 5 ans d'expérience dans l'univers du Développement Web et de la Programmation Java dans l'Industrie et la R&D. Il est souvent intervenu en méthodologie agile lors de ses différentes missions en tant que consultant IT et R&D sur des projets innovants et Développeur Web chez Carrefour, ERDF et Eutelsat.*

*Fin 2016, Google a dévoilé son assistant virtuel Google Home tant attendu. Au cœur du quotidien de la maison connectée, il est capable de répondre à toutes les demandes par commande vocale.*

Quelques développeurs PALO IT se sont lancé le défi de détourner l'usage initial d'un drone Parrot en lui intégrant la reconnaissance vocale ! Sur la base d'un cas client concret, nous explorerons les APIs de commandes vocales et d'objets connectés. Ce projet a été développé en mode Mob Programming et a permis d'aborder les technologies d'assistants virtuels par une approche sémantique pour interpréter le langage humain, analyser sa structure et son sens. Ce dernier a fait l'objet d'un BarCamp (https://www.youtube.com/watch?v=dxh80zClPTI) chez PALO IT : n'hésitez pas à visionner la vidéo !

## L'idée et l'état de l'art

L'aventure a commencé par une demande client : construire un objet connecté qui aurait pour but de fluidifier la relation entre les équipes et les clients. Il s'agissait d'un assistant virtuel qui prenait en charge les commandes et les intégrerait au CRM.

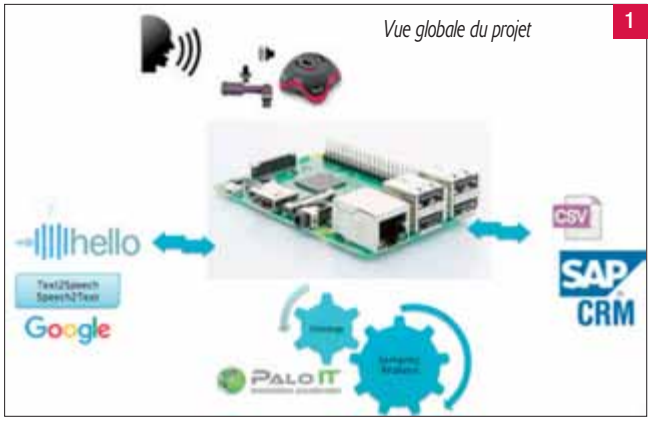
Après un tour d'horizon des possibilités existantes, nous nous sommes orientés vers une solution spécifique basée sur un composant de type Raspberry Pi connecté au service Google de reconnaissance vocale. Le tout serait embarqué dans un objet imprimé en 3D. [1]

En effet, Google et Amazon proposent des solutions brillantes qui ciblent le grand public avec leur "assistant vocal familial". En explorant un peu les conditions d'extension de ces produits, nous avons constaté que les contrats risquaient d'entraver la liberté de notre client. Soit avec des exigences particulières en termes de licences, soit avec un traitement sensible et non sécurisé concernant la confidentialité des données client. [2]

Avant de nous lancer, nous avons vérifié si d'autres personnes avaient déjà tenté l'expérience. Une première initiative R&D a été proposée à l'université de Vancouver, grâce à des tests des chercheurs qui étaient parvenus à intégrer des commandes vocales afin de faire voler ou atterrir un drone. Pour cela, il suffit simplement de s'adresser au drone par son identifiant avant de prononcer la commande souhaitée. En parallèle, plusieurs initiatives de ce genre ont commencé à émerger. Le X-Voice Drone propose quelques commandes vocales simples fonctionnant également avec une manette de contrôle.

***Notre idée était d'arriver à parler au drone comme à un chatbot d'une façon naturelle, sans être limité à quatre ou cinq commandes prédéfinies et préenregistrées.*** [3]

Comme l'idée est très prometteuse, nous avons décidé de réaliser un prototype. Nous avons choisi le drone "Jumping Sumo" de Parrot et nous avons commandé un Raspberry Pi avec lequel nous avons commencé à explorer différentes solutions pour avoir un prototype en quelques jours. C'est parti, l'aventure commence ! [4]




1 *Vue globale du projet*


3 *Projet des chercheurs de Vancouver*


2 *Assistants virtuels existants*

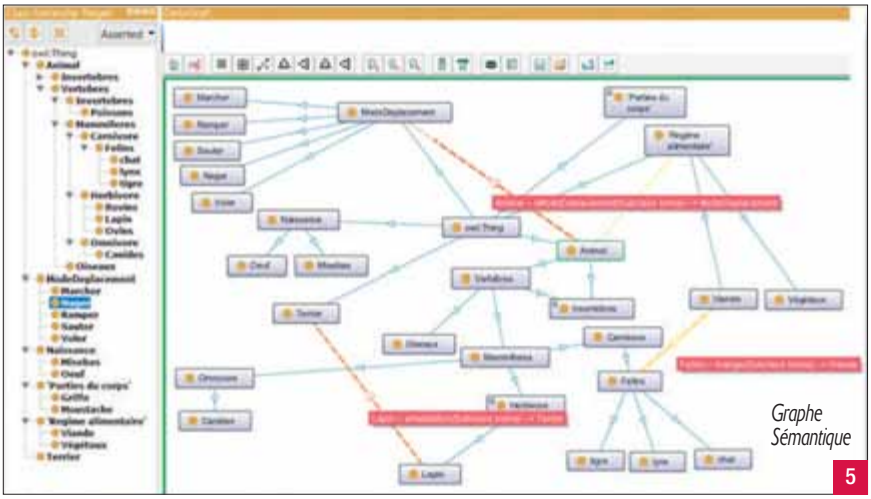
4 *L'aventure commence*

## La méthode

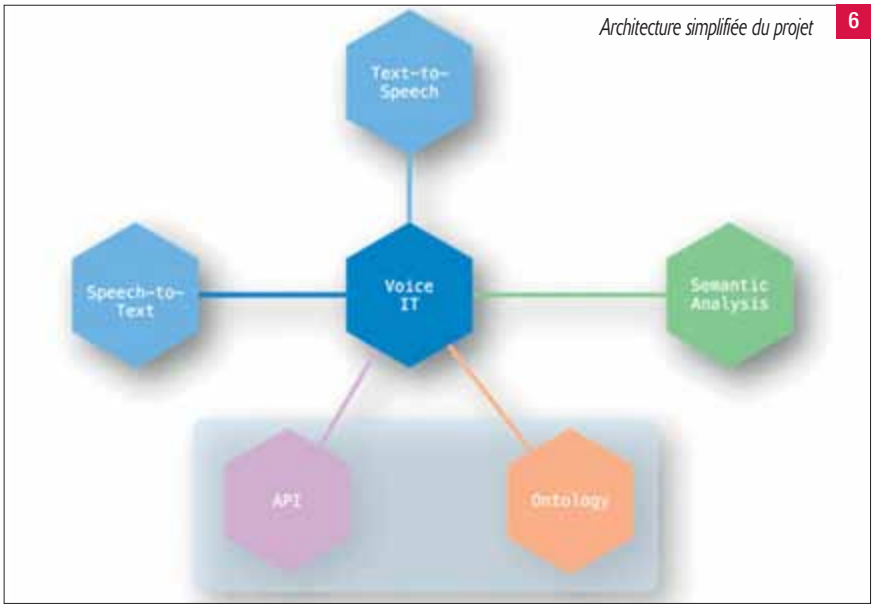
Pierre Dac disait *"Avec de la méthode et de la logique on peut arriver à tout aussi bien qu'à rien."*

Avec de la méthode et de la logique…

- Principes :
  - Nous allons à l'essentiel ;
  - L'équipe décide ensemble.
- Outils
  - Des User Stories ;
  - Un tableau Kanban ;
  - Un daily meeting ;
  - Des démos (dès qu'on peut, quand on peut).
- Repères
  - Une vision claire ;
  - De l'envie et du fun !



Graphe Sémantique 5




Architecture simplifiée du projet 6

## L'approche innovante

Nous devions procéder méthodologiquement et réaliser un produit qui fonctionne. Notre robot doit comprendre ce qu'on lui dit, pas uniquement obéir à quelques ordres prédéfinis. Il doit reconnaître le sens des mots et donc les analyser en se basant sur une approche sémantique.

La sémantique est une branche de la linguistique qui est à l'opposé de la branche qui se base sur la syntaxe. Elle étudie les signifiés : ce dont on parle et ce que l'on veut énoncer.

Pourquoi a-t-on besoin de la sémantique dans ce type de projet ? Elle va permettre de comprendre de quoi on parle. Elle permet de démêler les choses compliquées du langage comme les mots composés, les antonymes, les synonymes, etc. Dans le prototype, c'est ce qui va permettre au drone d'interpréter par exemple « tourne, mais pas à droite » ce qui donne en résultat une redirection à gauche grâce à l'interprétation sémantique. C'est cette capacité inédite qui est garante de l'innovation.

L'ontologie est un autre outil souvent utilisé par la sémantique. Elle permet de décrire la nature et les propriétés des objets du monde réel à partir de concepts et de relations entre ces concepts. En philosophie, l'ontologie est l'étude de l'être en tant qu'être. En science, c'est l'étude des propriétés générales de ce qui existe. C'est l'une des approches de modélisation qui permet d'inférer des faits à partir d'autres faits et surtout de donner du sens à une syntaxe.

Si nous allons un peu plus loin, une ontologie est un ensemble de concepts reliés entre eux. Ce genre de réseau de relations permet de faire ce qu'on appelle une inférence. C'est tout simplement une déduction à partir de conditions initiales.

Cette capacité d'émergence à partir des relations entre concepts est précieuse pour amener les machines à des comportements riches, proches de ce qu'on appelle l'intelligence. Avec des ontologies très complexes et riches en relations, le modèle serait analogue à un réseau de nœuds capable de générer des réponses à des interrogations sur l'ensemble de bases de connaissances. C'est à l'image du cerveau : plus il est grand et complexe, plus l'être vivant qui s'en sert, est intelligent. À partir d'un certain degré de complexité (nombre d'éléments et de relations) un modèle sémantique devient capable de faire émerger des connaissances inconnues jusqu'alors. [5]

## L'architecture

L'idée était d'avoir quelque chose de réutilisable, facile à intégrer dans différents métiers. En séparant les différents concepts, on reste agnostique par rapport au problème à résoudre : la seule chose à modifier pour traiter un nouveau client/API/Service, c'est l'ontologie et son mapping vers l'API finale.

Procédons étape par étape dans le processus d'interactions entre les différents composants :

- Un dispositif reconnaît la voix et la traduit vers du texte ;
- Notre application le reçoit et l'envoie vers le module du matching sémantique pour en extraire des actions et entités ;
- L'application essaye de matcher ces actions dans l'ontologie ;
- Si elle a reconnu une commande, on l'exécute avec l'API cible ;
- Le programme produit un feed-back utilisateur via text-to-speech ou un écran ;
- Dans le cas où l'application n'a pas compris, on peut commencer une 'conversation' avec l'utilisateur en demandant soit de répéter soit de donner plus de détails, exemple :
  - Utilisateur : "Tournez",
  - Robot : "Je peux tourner à gauche ou à droite, je fais quoi ?",
  - Utilisateur : "à droite !" [6]

7 CMU Sphinx
Project by Carnegie Mellon University

Nous avons commencé par explorer des solutions Open Source pour la reconnaissance vocale et l'analyse sémantique fonctionnant avec un Raspberry PI. Nous avons étudié les outils suivants : CMU (Carnegie Mellon Uni) Sphinx (Voice-to-Text) et Stanford NLP.

- Caractéristiques de CMU Sphinx : [7]
  - Le Pocketsphinx offre une version prête à utiliser sur Android (avec 20% de taux d'erreur parmi ~10 mille mots) ;
  - Il peut fonctionner en Offline ;
  - Le réglage (tuning) du modèle est assez compliqué ;
  - Le support de la langue française n'est pas assez complet, beaucoup moins que le support de l'anglais

Caractéristiques de Stanford NLP [8]

8

- Le support complet des modules que pour l'anglais, cependant nos commandes seront en français ;
- Stanford NLP n'a pas deux des éléments les plus importants pour nous : les Lemmes (*Lemmas*) et les *Named Entities*.
- Les *Lemmes* permettent d'extraire l'origine d'un mot depuis une conjugaison : en anglais : "was" -> "to be";
- Les *Named Entities* permettent de 'taguer' des mots, d'ajouter de la metadata, "yesterday" -> 26-10-2016.[9]

Il existe des outils Cloud permettant d'analyser et d'extraire des actions depuis du texte : Google, Wolfram sont les plus connus, mais il existe également des services comme api.ai et wit.ai. Tous ces services sont payants, mais l'analyse sémantique été bien au-delà de notre simple P.o.C. (Proof of Concept). Afin de bénéficier d'une analyse sémantique de qualité, il faut prendre en compte un coût d'implémentation assez important.

| ANNOTATOR | AR | ZH | EN | FR | DE | ES |
|---|---|---|---|---|---|---|
| Tokenize / Segment | ✔ | ✔ | ✔ | ✔ | | ✔ |
| Sentence Split | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Part of Speech | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Lemma | | | ✔ | | | |
| Named Entities | | ✔ | ✔ | | ✔ | ✔ |
| Constituency Parsing | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Dependency Parsing | | ✔ | ✔ | ✔ | ✔ | |
| Sentiment Analysis | | | ✔ | | | |
| Mention Detection | | ✔ | ✔ | | | |
| Coreference | | ✔ | ✔ | | | |
| Open IE | | | ✔ | | | |

9 *Supports des modules linguistiques*

## Choix de la plateforme du PoC : Tablette Android et Reconnaissance vocale.

Comme nous n'avions qu'un temps limité, nous avons repriorisé le backlog et décidé, au vu de ses différents avantages, de se lancer sur la plateforme Android d'une tablette :

- Speech-to-Text : il permet de télécharger les packs multilingues hors-ligne sur Android. Pour notre projet, il était nécessaire de pouvoir reconnaître la voix de manière offline, car le drone est connecté à la tablette via WiFi du drone. Dans la Raspberry, ça serait différent, car il y du WiFi ET de l'Ethernet ;
- Text-to-Speech : Non seulement on peut choisir la langue, mais on peut aussi choisir parmi différents types de voix (féminine/masculine, rapide/lente, etc.) ce qui facilite la personnalisation du produit pour ses utilisateurs (ceci peut être adapté aux capacités de l'utilisateur : enfant, adulte, etc.) et la communication avec ses interlocuteurs (ceci est très utile pour s'adapter au type du produit final : drone, robot, station fixe, application mobile / assistant virtuel, bot, etc.) :
- Écran + Micro + Haut-Parleurs intégrés ;
- Librairie GAST : encapsule des outils pour utiliser les capacités de détection d'événements sur les capteurs (dans notre cas le son) de la tablette Android. Il contient des exemples de codes et les algorithmes dont vous avez besoin pour utiliser correctement les capteurs pris en charge par Android.

## Le Drone

Parrot propose une API très complète avec laquelle on peut contrôler des mouvements, prendre des photos, transférer de la vidéo en temps réel, réaliser des animations, etc. Pour les modèles de drones plus avancés, l'API a la possibilité de définir un parcours via des coordonnées GPS, d'envoyer et de recevoir de l'audio, etc. De plus, l'API est écrite en C et propose des interfaces en Java, Unix et iOS. [10]


10 *Jumping Sumo Parrot*

## Où est l'innovation ?

Notre prototype utilise une interprétation sémantique et non juste syntaxique. Notre solution sait reconnaître un ensemble de mots et les interpréter dans un ensemble de concepts reconnus dans le domaine métier du client. Nous modélisons donc le champ sémantique de notre client pour que la reconnaissance soit vraiment ouverte et pertinente. Ceci peut être réalisé rapidement en adoptant un modèle sémantique déjà existant ; aujourd'hui il existe des modèles sémantiques génériques ou spécialisés pour plusieurs disciplines / activités (ex, médecine, tourisme, réservation en ligne, etc.) et qui sont même disponibles en libre-service sur des moteurs de recherche comme Google. On peut aussi les adopter et les améliorer pour les adapter. Il est également possible de construire son propre modèle.

L'avantage de cette approche est d'avoir une solution simple, ouverte, performante sans avoir besoin d'une phase d'apprentissage. En ce qui concerne la partie applicative, il n'y a pas de code spécifique au métier. À tout moment, on peut utiliser cette application pour un autre métier : il suffit alors de créer un nouveau lexique et de changer les appels d'API. L'innovation est là !

## Perspectives

Nos perspectives d'évolution sur ce projet sont les suivantes :

- Modulariser le code ;
- Analyse sémantique et extraction des actions/paramètres ;
- Mode conversation :
- Étendre l'ontologie et l'exploiter via Apache Jena ;
- Reconnaissance vocale.
- Intégrer tout ça dans un Raspberry Pi ;
- Utiliser la caméra du drone pour reconnaître des objets, couleurs, etc.

**Liens :**

http://blog.palo-it.com/2016/11/09/la-video-du-barcamp-ok-palo-presente-moi-ton-assistant-virtuel-est-en-ligne