



## Robot journalist "Le Monde"

---

Mickael Cianamea, Corentin De Mari and Thomas Dulcamara

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

January 5, 2022

# Robot journaliste "Le Monde"

Cianaméa Mickael <sup>1</sup>, De Mari Corentin <sup>2</sup>, Dulcamara Thomas <sup>2</sup>

Ceri Avignon, France

<sup>1</sup>mickael.cinamea@alumni.univ-avignon.fr, <sup>2</sup>corentin.de-mari@alumni.univ-avignon.fr,  
<sup>3</sup>thomas.dulcamara@alumni.univ-avignon.fr

## Extrait

Cet article a pour but de vous présenter un applicatif innovant dans le domaine de l'interaction entre un être humain et un robot, celui-ci s'occupe de la prise d'information dans l'actualité tel un journaliste. De la conception à l'évaluation du système nous parlerons du détail de l'architecture, en passant par les améliorations qui ont été apportées. Vous aurez la possibilité de comprendre comment nous avons procédé dans les démarches de développement, mais aussi quelles fonctionnalités nous avons choisi d'apporter à notre robot.

**Mot clés:** reconnaissance vocale, interaction humain-machine, robot journaliste

## 1. Introduction

Le principe de notre projet est de concevoir un applicatif innovant sous la forme d'un robot qui n'est autre que NAO développé par SoftBank Robotics, qui serait en capacité de vous fournir les principales actualités journalistiques.

L'intelligence de la machine et sa collecte d'information se base sur le journal français "Le Monde". Le finalité est de pouvoir entreprendre une communication humaine entre un robot et un utilisateur. Cette automate interagira avec ce dernier en lui proposant de lui lire les dernières actualités selon des catégories ainsi que leurs contenu.

## 2. Architecture de l'application

L'application s'articule en 3 services bien distinct dont un serveur de dialogue, un service de retranscription vocal et un module de programmation du robot. Voici comment tous ces services s'imbrique pour donner le résultat suivant :

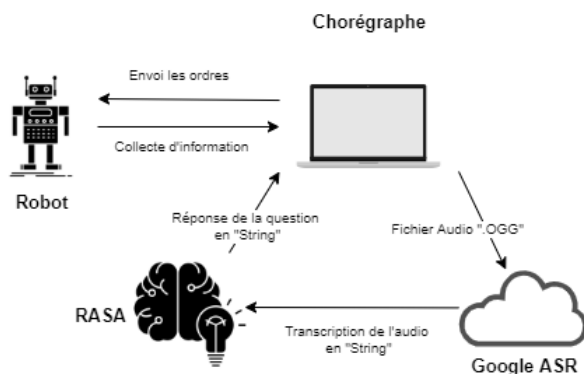


Figure 1: Schéma de communication du système

### 2.1. Serveur de dialogue

Le premier service est un serveur de dialogue de type chat bot. Afin de parvenir à nos fins, nous avons utilisé RASA X qui est la solution graphique adapté pour la conception d'un chat bot. Ce service nous a été proposé par l'équipe enseignante et nous avons fait le choix d'apprendre sa logique et son fonctionnement pour construire un système robuste.

Les choses essentielles à comprendre concernant RASA X, sont les différentes nomenclatures des items de son système.

Le premier item est l'"Intent", ce qui correspond aux phrases pouvant être fournies à notre chat bot par l'utilisateur. Nous lui fournissons donc des phrases types, comme dans notre cas "Donne-moi les informations" ou "Donne-moi le détail de l'article 3". Bien entendu RASA est intelligent, est si la phrase fournie ne correspond pas exactement à un modèle, il trouve la phrase type la plus proche grâce au coefficient d'approximation par la suite, nous avons la possibilité d'ajouter la phrase précédente en phrase type avec une interface adaptée.

L'"utter" est le deuxième item répertoriant les différentes réponses possibles de notre chat bot. Nous pouvons définir une multitude de phrase enregistrées pour une seule réponse. C'est le système RASA qui est en charge de choisir la réponse à fournir à l'utilisateur.

L'"action" est le troisième item et représente les actions métier propre à notre robot. Dans notre cas, nos actions permettent de récolter les données d'information à partir de flux RSS du journal le monde. Ces actions permettent de donner des réponses plus complexes que des réponses de type "utter". Ces actions ne sont rien d'autre que des méthodes python au sein du fichier action.py de RASA X.

Le dernier item est la "story". C'est ce dernier item qui est le chef d'orchestre pour faire fonctionner la logique de notre chat bot. La story prend l'"Intent" de l'utilisateur et y associe un ensemble d'étapes successives constituées d'"utter", d'"action" et de "checkpoint". Les "checkpoints" sont des points de passage d'un état bien précis. Cela permet de faire des boucles et des enchaînements de "story" sans pour autant écrire tous les parcours utilisateurs.

Afin d'interagir avec le chat bot, nous avons deux moyens. Le premier très utile lors des phases de test de notre serveur de dialogue est une interface web et le second est un webhook très utile pour la conception de système inter-connecté. C'est donc sur ce deuxième moyen que nous nous tournons pour notre application.

## 2.2. Programmation du robot

Le robot pris en étude pour cette application est le robot Nao par le constructeur SoftBank Robotics. Le constructeur édite son logiciel de gestion de robot se nommant Chorégraphe. La programmation de celui-ci, c'est donc réalisé avec le logiciel Chorégraphe permettant de disposer de manière graphique des blocs prêt à l'emploi programmé en python, permettant des actions pouvant être lié les un aux autres dans un ordre précis afin d'enchaîner différentes actions.

Dans un premier temps le robot va vérifier sa position et s'il n'est pas debout, il va demander à l'utilisateur d'attendre et va se lever pour ensuite vous saluer et dire qu'il est à votre disposition. Ensuite la communication entre le robot, l'ASR, et rasa débutera. À partir de ce moment les réponses du robot seront contrôlés par rasa, où nous aurons une multiple de dialogue pour interagir avec lui.

## 2.3. Connexion robot-serveurs

Une fois le comportement programmé grâce à Chorégraphe, nous avons dû faire le choix d'un logiciel de reconnaissance et synthèse de parole plus complet contrairement au comportement de base qui reste assez limité. Notre choix c'est porté sur la solution en ligne de Google qui se nomme "Google ASR". L'utilisation de celui-ci va fonctionner de la manière suivante; notre serveur attendra un fichier audio de type ".OGG" fourni par le robot, puis nous l'enverrons via une requête de type POST via l'API REST de Google. Une fois le traitement de l'ASR effectué, nous obtenons la réponse de l'API, qui nous répond via un objet "String".

Cet objet sera ensuite collecté par Chorégraphe pour le distribuer à notre serveur RASA X. Ce deuxième serveur, permet de gérer deux choses: la première étant l'interface de RASA et la deuxième est l'intégration de l'API du journal "Le Monde".

## 3. Tests et évaluation du système

Concernant l'évaluation de notre système, nous avons dans un premier temps procéder à des tests internes à l'équipe en s'assurant que notre système fonctionne correctement dans son champ d'action. Dans un second, temps nous avons fait tester notre création à un autre groupe d'étudiants afin de déceler d'éventuels erreurs.

Lors des tests réalisés par notre équipe, nous avons fait face à de multiple problèmes. Les premiers problèmes rencontrés étaient les installations des différents services sur nos environnement de travail. Afin de résoudre ces problèmes, nous avons dû trouver des versions compatibles, selon les machines utilisées. Dans notre cas nous avons dû installer une version spécifique de RASA X et chorégraphe sous des environnements Windows.

Le second problème majeur rencontré était la gestion de la transmission du fichier audio vers Google ASR à partir du robot. Nous avons voulu comprendre en détail son fonctionnement et nous avons donc modifié des paramètres afin de le mettre à notre convenance, mais lors de ces opérations, la transmission du fichier ne s'est plus faite. Afin de résoudre ce problème, nous avons récupéré les configurations de base par défaut.

Une fois la première version de notre système finalisé, nous avons réussi à le faire fonctionner, que ce soit la partie serveur de dialogue (RASA X), le service de retranscription (Google ASR) ou même le robot et l'interaction entre toutes ces parties. Les résultats que nous avons obtenues sont égaux

à nos attentes, le système à une bonne compréhension de notre parole et en retour nous avons aussi un très bon retour du robot. Nous avons pu constater que l'interaction globale est fluide, ressemblante à une conversation humaine et compréhensible.

Le groupe à qui nous avons fait tester notre système, nous a fait un très bon retour, ils ont trouvé l'interaction avec le robot très humaine et fluide, ainsi que des réponses cohérentes à leur demande.

## 4. Tâches du robot

### 4.1. Tâches originales

Dans sa première version, notre application se résume aux stories incluant le flux RSS RASA que voici :

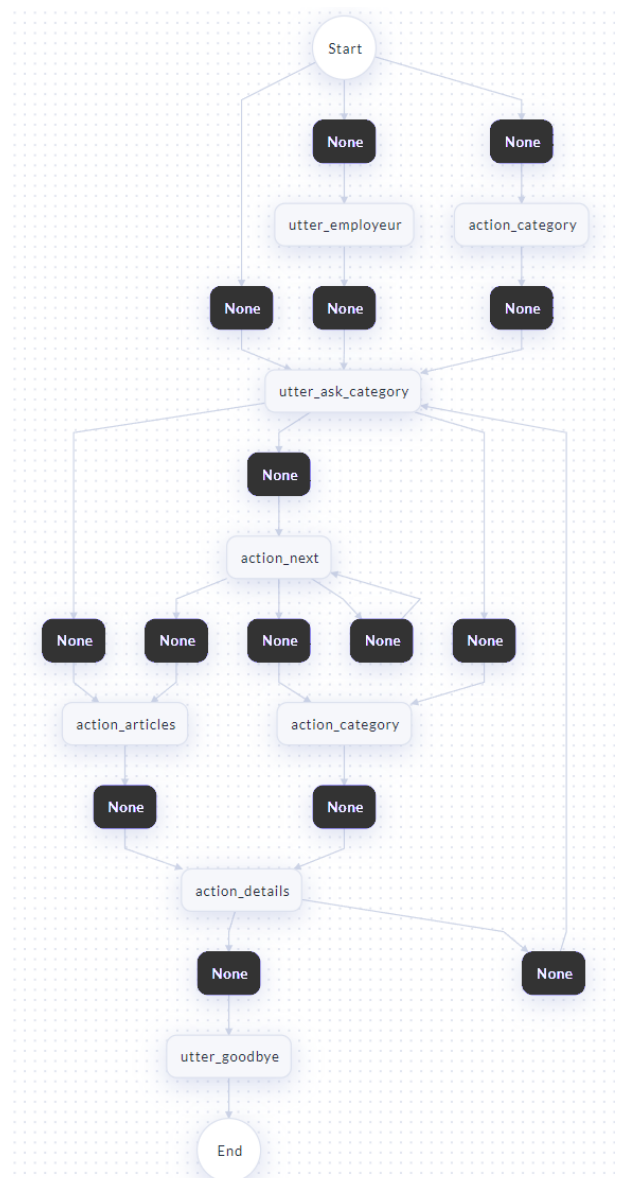


Figure 2: Schéma des stories RASA

Les blocs "None" noir correspondent à des "intents" qui sont appelés en fonction de la réponse de l'utilisateur.

Dans l'idée générale, voici l'une des histoires d'un utilisateur.

- Utilisateur : "Donne moi des informations."
- Bot : "Quelle catégorie souhaitez-vous ?"
- Utilisateur : "Quelles sont les catégories ?"
- Bot : "Les catégories disponibles sont : Politique, sport, ..."
- Utilisateur : "Catégories sport"
- Bot : "Les articles sont : Article 1: Lorem ..., Article 2. Lorem..., ..."
- Utilisateur : "Détail de l'article 1"
- Bot : "Lorem ipsum ..."
- Utilisateur : "Au revoir merci"
- Bot : "Au revoir mon ami"

En dehors des tâches fonctionnelles incluant l'utilisation du flux RSS du journal "Le Monde", vous avez la possibilité d'interagir avec le robot via le toucher sur le haut de son crâne, mais aussi le voir faire quelques gestuelles. Si vous lui demandez gentiment qui est son employeur, peut-être qu'il vous répondra ? Qui sait.

#### **4.2. Tâches évolutives**

Afin de produire une nouvelle version, nous avons imaginé des nouvelles fonctionnalités additionnelles.

La première idée est de rajouter de nouveaux médias tel que BFM ou autre afin d'obtenir des divergences d'opinions des informations et d'avoir un avis critique sur ce que nous entendons.

Une seconde nouveauté sera la personnalisation des informations rapportées par de la reconnaissance faciale ou via des données personnelles telles que l'adresse mail.

### **5. Conclusions**

En conclusion, lors de la mise en place des différentes parties nous avons rencontré de nombreux problèmes comme les installations des différents services et par la suite, entre les communications robot-Google ASR.

Au final nous avons un système qui est entièrement fonctionnel et nous arrivons à avoir une interaction "humaine" avec le robot entre la programmation du robot et les réponses renvoyées par rasa. Nous avons conçu un robot pouvons-nous donner des informations journalistiques en temps réel.

Il a été très intéressant de travailler sur ce projet et de pouvoir traiter les différents aspects afin d'arriver à concevoir un robot imitant une conversation humaine. Nous avons pu entrevoir toutes les possibilités d'innovations avec ces technologies et une manière de penser à la simplification du quotidien.