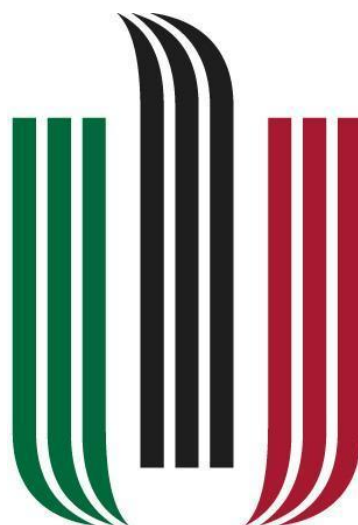


Wydział Informatyki, Elektroniki i Telekomunikacji

Katedra Informatyki



AGH

Technologie Obiektowe i Komponentowe
Zdalne centrum komend głosowych dla robota
NAO

Michał Torba

2014 r.

1 Wstęp

Każdy robot, obojętnie od stopnia zaawansowania jest ograniczony przez dostępne dla niego zasoby. Jego program ma skończony poziom skomplikowania którego nie będzie w stanie przeskoczyć – chyba, że intensywna obliczeniowo część zostanie wykonana na zewnętrznych komponentach. Jedną z rzeczy którą możnaby w ten sposób wydzielić jest komponent komunikacji głosowej – parsowanie mowy naturalnej i przekładanie jej na logikę zachowań robota jest monumentalnym wyzwaniem zarówno dla oprogramowania jak i sprzętu, naturalnym więc jest byłoby wydzielenie go poza robota właściwego.

2 Założenia projektu

2.1 Cel

Celem projektu jest stworzenie systemu pozwalającego na zdalne przechowywanie logiki obsługującej komunikację głosową z robotem i wykorzystanie jej. Robot odczytywałby mowę ludzką korzystając z wbudowanego w niego parsera mowy, przesyłał komendę do serwera komend a w odpowiedzi dostawał przygotowany zestaw instrukcji do wykonania.

2.2 Artefakty

Wynikiem projektu będą dwa artefakty:

- Aplikacja kliencka
- Aplikacja serwerowa

2.3 Środowisko

Aplikacja kliencka musi działać w ramach robota NAO z funkcjonującym mikrofonem i wbudowanym parserem mowy. Aplikacja serwerowa w założeniu ma działać na komputerach z systemem operacyjnym Ubuntu 13.10

3 Wymagania

3.1 Funkcjonalne

1. Aplikacja serwerowa jest w stanie obsłużyć (znaczy – w oparciu o wejściową komendę wyprodukować instrukcje dla robota) dwie komendy głosowe
2. Robot odczytuje komendy głosowe użytkownika
3. Robot wykonuje komendę użytkownika zgodnie z jej implementacją na serwerze komend.

3.2 Niefunkcjonalne

1. Możliwe jest dodawanie dodatkowej funkcjonalności (obsługę nowych komend) bez ingerencji w kod aplikacji klienckiej.

4 Sposób rozwiązania problemu

Interfejs parsowania mowy wbudowany w NAO niestety nie pozwala na rozpoznanie dowolnego słowa. Z tego powodu działanie systemu wygląda następująco

- Klient wysyła do serwera prośbę o listę dostępnych komend (interfejs RESTowy)
- Serwer odpowiada wysyłając listę komend posiadających implementację
- Klient wykorzystując interfejs parsowania mowy czeka aż użytkownik wypowie któreś z obsługiwanych słów
- Klient po rozpoznaniu takiego słowa wysyła je do serwera
- Serwer w odpowiedzi przesyła w jakiejś formie (prawdopodobnie odpowiednia reprezentacja tekstowa) komendy dla robota do wykonania
- Klient wykorzystuje interfejs robota i wykonuje komendy zawarte w wiadomości zwrotnej

5 Kamienie milowe

1. **Komunikacja** – napisanie działającej na robocie aplikacji pozwalającej na połączenie się z serwerem.
2. **Wymiana danych i integracja** – W pełni ukończony interfejs komunikacji pomiędzy klientem i serwerem, łącznie z protokołem komunikacji dotyczącym dostępnych dla części serwerowej komend robota. Obsługa przykładowej komendy po stronie serwerowej, wykorzystanie API robota w celu odczytania komendy głosowej oraz wykonania instrukcji zawartych w odpowiedzi serwera.

6 Analiza ryzyka

Nieznajomość środowiska pracy – W chwili obecnej ciężko powiedzieć jakie ograniczenia niesie ze sobą platforma jaką jest robot NAO. Być może napisanie aplikacji typu klient-serwer będzie tak proste jak wszędzie. Być może okaże się niemożliwe przy postawionych założeniach. Być może będzie możliwe ale z jej poziomu nie będzie można odwołać się do API sterującego działaniem robota. W każdym wypadku będzie wymagało dużo pracy eksploracyjnej

Środowisko do testów – Nie mam pewności, czy całe niezbędne środowisko da się zasymulować używając komputera osobistego. Co z odczytywaniem komend głosowych? Alternatywą jest konieczność testowania na rzeczywistym sprzęcie co budzi uzasadnione obawy i powoduje znaczące trudności natury dostępowej

Język pracy – Wszystko wskazuje na to, że projektu nie będzie można zrealizować korzystając z języka Java. Wykorzystanie Pythona lub C++ powoduje zaś dodatkowe problemy związane z brakiem doświadczenia w programowaniu w tych językach.