

Robot Operating System

Adam Wolniakowski

Politechnika Białostocka

30 października, 2014

Co to jest ROS?

ROS (*Robot Operating System*) to open-source'owy, meta-system operacyjny, przeznaczony do zastosowań w robotyce. Zapewnia to, czego można oczekiwać od systemu operacyjnego:

- ▶ abstrakcję hardware
- ▶ kontrolę urządzeń na niskim poziomie
- ▶ przekazywanie danych między procesami
- ▶ komunikacja między różnymi hostami
- ▶ ...

Reinventing the wheel

How Robotics Research Keeps....

Re-Inventing the Wheel

First, someone publishes...



...and they write code that barely works but lets them publish...



...a paper with a proof-of-concept robot.



This prompts another lab to try to build on this result...



But inevitably, time runs out...



...but they can't get any details on the software used to make it work...



...and countless sleepless nights are spent writing code from scratch.



So, a grandiose plan is formed to write a new software API...



...and all the code used by previous lab members is a mess.

Willow Garage

PHD COMICS visits Willow Garage

WILLOW GARAGE IS A START-UP ROBOTICS COMPANY IN THE HEART OF SILICON VALLEY.

AN ENTIRE COMPANY DEDICATED TO ONE GOAL: TO MAKE PERSONAL ROBOTICS A REALITY.

PING-PONG TABLE

FREE FOOD! ROBOTS!

THEIR STRATEGY TO ACCELERATE THE RATE OF ROBOTICS RESEARCH EVERYWHERE.

IT'S ONLY A MATTER OF TIME WE WANT TO MAKE IT HAPPEN FASTER.

STEVE COUGAR, CEO

HOW ARE THEY DOING THIS?

FIRST, WILLOW GARAGE ANNOUNCED THEY WOULD GIVE AWAY TEN GRATE-FREE ART PERSONAL ROBOTS (THE PR2) TO LABS AND INSTITUTIONS AROUND THE GLOBE.

THE GOAL IS FOR EVERYONE TO HAVE A COMMON PLATFORM ON WHICH TO SHARE AND COMPARE RESULTS.

FREE ROBOTS? CAN I HAVE ONE?

UH, NO.

THEY ALSO STARTED A FREE ONLINE OPEN-SOURCE ROBOT PLATFORM CALLED ROS (ROBOT OPERATING SYSTEM).

ANYONE CAN DOWNLOAD THE CODE, IMPROVE IT, AND SHARE THEIR INNOVATIONS.

YOU CAN USE IT ON ANY ROBOT, NOT JUST THE PR2!

DRIVERS

NAVIGATION ALGORITHMS

DEVELOPMENT TOOLS

A TYPICAL GRAD STUDENT SPENDS TONS OF THEIR TIME RE-INVENTING THE WHEEL.

BY BRINGING THE COMMUNITY TOGETHER AND PROVIDING TOOLS FOR SHARING AND BUILDING ON EACH OTHER'S WORK, WE'LL BE ABLE TO ACCOMPLISH SO MUCH MORE.

WE WANT TO TAKE THE GRUBT WORK OUT OF GRAD WORK.

OK, WHAT IF I DRAW A COMIC ABOUT YOU?

CAN I GET A ROBOT THEN?

KEENAN AND MIKE CO-DIRECTORS

Why?

Dlaczego warto zastanowić się nad użyciem ROS-a?

- ▶ Nie ma sensu implementować wszystkiego od zera
- ▶ Gotowe sterowniki dla wielu urządzeń
- ▶ Bardzo aktywna społeczność
- ▶ ROS jest fajny

ROS integruje kilka interesujących rzeczy:

- ▶ OpenCV
- ▶ PCL
- ▶ SLAM
- ▶ Sterowniki: Kinect, IMU, kamery, GPS, ...
- ▶ Planowanie trajektorii
- ▶ Symulatory

POJĘCIA

Poniżej spróbujemy wyjaśnić kilka podstawowych pojęć:

- ▶ node
- ▶ message
- ▶ topic
- ▶ service

Struktura ROS-a

ROS jest zorientowany na komponenty.

Komponent ROS-a nazywamy *node*. Komponenty komunikują się między sobą przy pomocy strumieni danych (*topics*), lub wywołując usługi (*services*).

Node

Node to komponent systemu. Każdy z nich może być uruchomiony oddzielnie, w dowolnym momencie, na dowolnym hoście. *Node*-y posiadają wejścia i wyjścia.

Komponentem może być na przykład:

- ▶ sterownik robota
- ▶ kamera
- ▶ program odpowiedzialny za przetwarzanie danych
- ▶ archiwum
- ▶ ...

Message

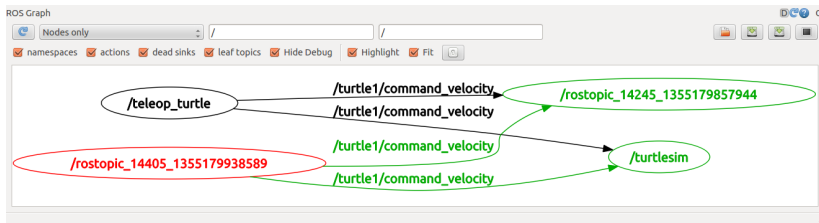
Message to pojedyncza ramka informacji przesyłana pomiędzy komponentami. ROS wyposażony jest w bogatą bibliotekę gotowych wiadomości (np. PCL, transformacje, macierze, obrazy, typy standardowe, ...). Wiadomości można też łatwo zdefiniować samemu.

Format wiadomości

```
#komentarz  
float64 myDouble  
string myString  
float64[] myArrayOfDouble
```

Topic

Topic jest strumieniem danych przekazywanych pomiędzy komponentami. Do jednego *topic*-u mogą jednocześnie publikować i subskrybować dane różne komponenty.
Topic to zazwyczaj dane publikowane regularnie.



Service

Service umożliwia wywołanie pewnej usługi, bądź komunikację asynchroniczną. Usługi można również definiować samodzielnie (format jest podobny do wiadomości).

MyService.srv

```
#wejście  
float64 dzielna  
float64 dzielnik  
---  
#wyjście  
float64 iloraz
```

Języki programowania

ROS sprawnie obsługuje następujące języki programowania:

- ▶ C / C++
- ▶ Python
- ▶ Java

Dodatkowo, można wykorzystać języki skryptowe (np. LUA), opisowe (XML), itd.

Współpraca z Matlabem i Simulinkiem

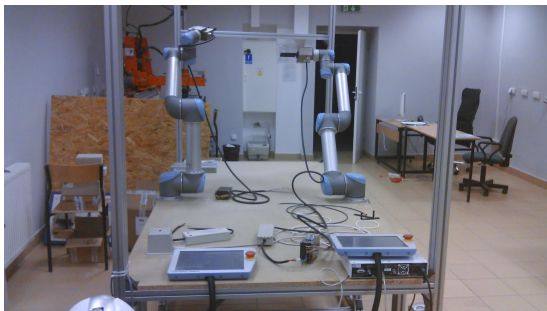
Można znaleźć toolboxy, które umożliwiają wykorzystanie systemu ROS z poziomu Matlaba, np. ROS IO toolbox.

Implementując odpowiednie funkcje MEX, można stworzyć bloki w Simulinku umożliwiające komunikację z systemem. Można w ten sposób zrealizować zapis danych i sterowanie w czasie rzeczywistym, budując odpowiedni model w Simulinku.

Stanowisko 'Bender'

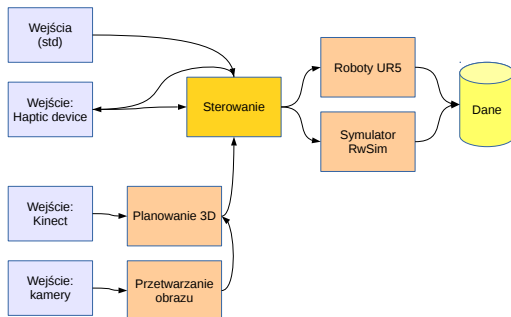
Stanowisko badawcze w sali 608 '*Na antresoli*' zawiera m.in. roboty UR5, kamery, TrakStar, chwytaki, ...

Docelowo ma być kopią stanowiska *MARVIN* z uniwersytetu SDU w Odense.



Struktura

Stanowisko *Bender* będzie obsługiwane przez system ROS. Dla każdego z urządzeń zostanie zaimplementowany komponent ROS, co umożliwi łatwą obsługę stanowiska zarówno na miejscu, jak i zdalnie.



Koniec

Dziękuję! Nareszcie!