Wstęp

Witamy! Jest to książka o systemach wielorobotowych. Dlaczego? Ponieważ to przyszłość!

Roboty stają się coraz tańsze, bardziej sprawne i użyteczne w wielu scenariuszach "prawdziwego życia". W rezultacie widzimy coraz więcej robotów, które muszą dzielić przestrzeń i współpracować ze sobą w celu wykonania zadań.

W tej książce przedstawimy **Robot Operating System 2 (ROS 2)** oraz **Robot Middleware Framework (RMF)**, który bazuje na ROS 2 i ma na celu uproszczenie tworzenia oraz obsługi złożonych systemów wielorobotowych. Ten rozdział opisuje motywacje i cele systemu ROS 2 oraz RMF w zakresie integracji wielu robotów.

ROS 2

Robot Operating System (ROS) to zestaw bibliotek oprogramowania i narzędzi do budowania aplikacji robotycznych. Od sterowników po najnowocześniejsze algorytmy oraz potężne narzędzia deweloperskie – ROS posiada wszystko, czego potrzebujesz do swojego kolejnego projektu robotycznego. A wszystko to jest oprogramowaniem typu **open source**.

Odkąd ROS powstał w 2007 roku, wiele zmieniło się w społeczności robotyki.

- **ROS 1** pierwotnie powstał jako środowisko programistyczne dla robota **PR2** firmy Willow Garage, wysokowydajnej mobilnej platformy manipulacyjnej przeznaczonej do zaawansowanych badań i rozwoju.

- Pierwotnym celem ROS było dostarczenie narzędzi programowych potrzebnych użytkownikom do podejmowania nowatorskich projektów badawczych z tym robotem.

- Jednocześnie zespół deweloperski wiedział, że PR2 nie będzie jedynym ani najważniejszym robotem na świecie, dlatego chcieli, aby ROS 1 był użyteczny także dla innych maszyn.

- Główny nacisk połączono z definiowaniem poziomów abstrakcji (zazwyczaj poprzez interfejsy komunikatów), co pozwalało na ponowne wykorzystanie oprogramowania w innych miejscach.

ROS 1 nie tylko spełnił wymagania PR2, ale okazał się przydatny w zaskakująco szerokiej gamie robotów. Obejmowało to roboty podobne do PR2, ale także roboty kołowe o różnych rozmiarach, humanoidalne roboty kroczące, ramiona przemysłowe, pojazdy terenowe (w tym samochody autonomiczne), drony, jednostki nawodne i inne. Adopcja ROS 1 wykroczyła poza społeczność akademicką, która była początkowym celem. Na rynek zaczęły trafiać produkty oparte na ROS 1, w tym roboty produkcyjne, rolnicze i komercyjne roboty sprzątające. Agencje rządowe również zaczęły przyglądać się ROS pod kątem zastosowania w systemach terenowych; **NASA**, na przykład, przewidywała uruchomienie ROS na Robonaut 2 wysłanym na Międzynarodową Stację Kosmiczną.

Wszystkie te zastosowania rozwinęły platformę ROS w nieoczekiwany sposób. Choć system radził sobie dobrze, zespół ROS 1 uznał, że może lepiej sprostać potrzebom szerszej społeczności, stawiając czoła nowym przypadkom użycia. **Tak narodził się ROS 2**.

Pierwotnym celem projektu ROS 2 było dostosowanie się do zmieniającego się krajobrazu, wykorzystanie tego, co było świetne w ROS 1 i poprawienie tego, co nie działało. Chciano jednak zachować ROS 1 w istniejącej formie, aby mógł nadal działać bez wpływu ze strony nowej wersji. Dlatego ROS 2 zbudowano jako równoległy zestaw pakietów, które można instalować obok ROS 1 i które mogą z nim współdziałać (na przykład poprzez mosty komunikatów).

Robotics Middleware Framework (RMF)

Wyobraź sobie przez chwilę dowolny duży budynek: centrum handlowe, kompleks mieszkaniowy, uniwersytet, miejsce pracy, lotnisko, szpital lub hotel. Czy wewnątrz budynku dostarczane są towary? Czy podłoga jest regularnie sprzątana?. Dla większości budynków odpowiedź brzmi "tak".

Zastanówmy się, co się dzieje, gdy te zadania zaczynają wykonywać roboty. Na dzisiejszym rynku można kupić doskonałe roboty dostawcze, jak i sprzątające. Co jednak, jeśli podłoga jest sprzątana w tym samym czasie, gdy w budynku dostarczane są przedmioty?. Taka sytuacja jest trywialna, gdy zadania wykonują ludzie: szybkie spojrzenie między dostawcą a sprzątaczem wystarczy, by osiągnąć kompromis. Jedna lub obie osoby znajdą sposób, by nieco zmienić czas wykonania swojego zadania.

Niestety, roboty nie są tak zdolne jak ludzie w zakresie abstrakcyjnego rozumowania, planowania i nieformalnej komunikacji!. RMF ma pomagać w unikaniu takich scenariuszy.

- Jeśli wszystkie roboty pochodzą od tego samego producenta (system **single-vendor**), będą one wiedzieć o swoim istnieniu i unikać konfliktów.

- Jednak systemy wielonarodowe i wielodostawcze (**multi-vendor**) pozostają problemem otwartym, a spodziewamy się, że w przyszłości będą one normą w dużych obiektach.

Aby zaradzić tej sytuacji, RMF dostarcza zestaw konwencji, narzędzi i implementacji oprogramowania, umożliwiając różnym flotom robotów współpracę ze sobą oraz z infrastrukturą budynku, taką jak windy, drzwi czy korytarze. Bez ram współpracy dla robotyki wielu dostawców, operatorzy budynków ponoszą ryzyko zmuszenia do korzystania z tylko jednego dostawcy, co generuje koszty integracji i ogranicza wybór przyszłych rozwiązań. Istnieje również konflikt o wspólne zasoby: windy, korytarze, przepustowość sieci, ładowarki czy zasoby ludzkie (personel IT). Te problemy były fundamentalną motywacją dla powstania RMF.

W scenariuszu "sprzątanie i dostawa", RMF może działać jako kontroler ruchu, pomagając robotom negocjować sposób wykonania obu zadań w zależności od ich priorytetu. Jeśli sprzątanie jest pilne (np. wyciek), RMF może skierować robota dostawczego inną drogą. Jeśli krytyczny czasowo jest transport, RMF nakaże robotowi sprzątającemu przerwanie pracy

i usunięcie się z drogi. Wyzwanie polega na tym, by te rozwiązania były generyczne dla wielu scenariuszy i "odporne na przyszłość" dla nieznanych jeszcze robotów i zastosowań.

Czym jest RMF?

Narzędzia i funkcje RMF:

- **Unified Robot Schedule:** Zunifikowany harmonogram robotów.
- **Robot Traffic Negotiation:** Negocjacje ruchu robotów.
- **Task Dispatch Auctions:** Aukcje przydzielania zadań.
- **Traffic Editor:** GUI do tworzenia i opisywania planów pięter na potrzeby RMF i symulacji w *Gazebo*.
- **Free Fleet:** Otwarty system zarządzania flotą dla deweloperów, którzy nie posiadają własnego menedżera floty.
- **RMF Schedule Visualizer:** Narzędzie oparte na rviz do wizualizacji i kontroli *rmf_core*.
- **RMF Web UI (rmf-web):** Aplikacja internetowa oferująca przyjazny dla operatora panel sterowania i wizualizację systemu RoMi-H.
- **RMF Simulation:** Wtyczki do symulacji RMF w środowiskach *Gazebo* i *Ignition*.

Wszystkie te implementacje są typu **open-source**, dostępne do wglądu i dostosowania. Projekt ten jest wspierany przez rząd Singapuru, w tym Ministerstwo Zdrowia (MOH) oraz Narodowy Program Robotyki (NRP).

