



Politechnika Wrocławska



Katedra Cybernetyki i Robotyki

Wprowadzenie do systemów rozproszonych

Mariusz Janiak
p. 331 C-3, 71 320 26 44

© 2017 Mariusz Janiak
All Rights Reserved



Zawartość wykładu

- Wprowadzenie do systemów rozproszonych
- Podejście zorientowane na komponenty w projektowaniu rozproszonych systemów sterowania
- Protokoły komunikacji
- Struktura ramowa OROCOS
- Struktura ramowa ROS
- Narzędzia i biblioteki programistyczne



Course credit terms

- Do zaliczenia kursu wymagane jest zdobycie zaliczenia ze wszystkich form towarzyszących: wykład (**F1**) i laboratorium (**F2**)
- Zaliczenie wykładu (**F1**)
 - kolokwium, próg 51%,
 - minitesty „Wnukówki” – podczas zajęć, 1-2 pytania, 3-5min, punktacja 0-3 (0 – nieobecność, 1 – pusta kartka, 2,3 – odpowiedź na pytania), ocena:
 - poniżej 60% konieczny egzamin,
 - 4.0 (db) od 60%,
 - 4.5 (db+) od 73,33%,
 - 5.0 (bdb) od 86,66%.
- Zaliczenie laboratorium (**F2**) wymaga aktywnego uczestnictwa w zajęciach laboratoryjnych: obecność obowiązkowa, zaliczenie wszystkich ćwiczeń, przygotowanie do zajęć, samokształcenie, próg 51%,
- Ocena końcowa

$$P = 0.4 * F1 + 0.6 * F2$$



Literatura

- Ch. Gerber, H.M. Hanisch, S. Ebbinghaus, *From IEC 61131 to IEC 61499 for Distributed Systems: A Case Study*, EURASIP Journal on Embedded Systems, 2007
- R. Simmons, D. Kortencamp, D. Brugali, *Robotics Systems Architectures and Programming*, Handbook of Robotics II-ed., Springer 2013
- A. J. A. Wang, K. Qian, *Component-Oriented Programming*, John Wiley & Sons, Inc., 2005
- D. Brugali, P. Scandurra, *Component-based robotic engineering (Part I)*, in Robotics & Automation Magazine, IEEE , vol.16, no.4, pp.84–96, December 2009
- D. Brugali, A. Shakhimardanov, *Component-Based Robotic Engineering (Part II)*, in Robotics & Automation Magazine, IEEE , vol.17, no.1, pp.100-112, March 2010



Pomoce dydaktyczne

- BRICS (<http://www.best-of-robotics.org>)
- OROCOS (www.orocos.org)
- ROS (www.ros.org)
- GAZEBO (<http://gazebo-sim.org>)
- VREP (<http://www.coppeliarobotics.com>)
- ACADO Toolkit (www.acadotoolkit.org)
- CasADi (<https://github.com/casadi/casadi/wiki>)
- Sundials (<https://computation.llnl.gov/casc/sundials>)
- Eigen (<http://eigen.tuxfamily.org>)
- ZeroMQ (<http://zeromq.org>)
- Protocol Buffers
(<https://developers.google.com/protocol-buffers>)



Wprowadzenie do systemów rozproszonych

Spróbujmy odpowiedzieć na pytania ...

- Czy trudno jest zbudować złożony system robotyczny?
- Jak uniknąć ponownego wynajdowania koła?
- Czy istnieje metodologia wspomagająca ten proces?
- Jakiego rodzaju narzędzia powinny zostać użyte?



Wprowadzenie do systemów rozproszonych

Oprogramowanie złożonych systemów robotycznych jest zazwyczaj

- wbudowane (embedded),
- współbieżne,
- czasu-rzeczywistego,
- rozproszone,
- i przetwarza duże ilości danych,

przy czym musi gwarantować

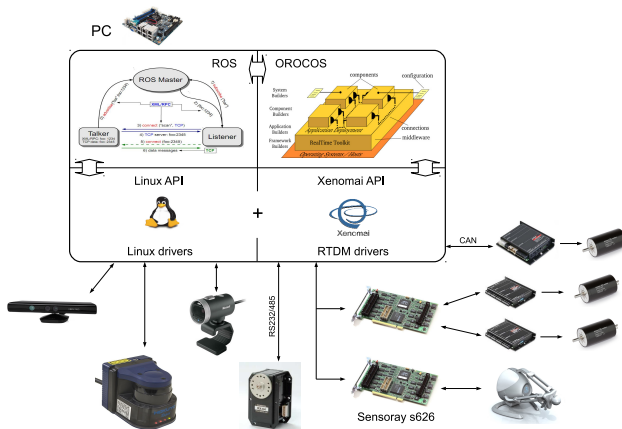
- bezpieczeństwo,
- niezawodność,
- odporność na uszkodzenia.¹

¹D. Brugali, P. Scandurra, *Component-based robotic engineering (Part I)*, in Robotics & Automation Magazine, IEEE , vol.16, no.4, pp.84–96, December 2009



Wprowadzenie do systemów rozproszonych

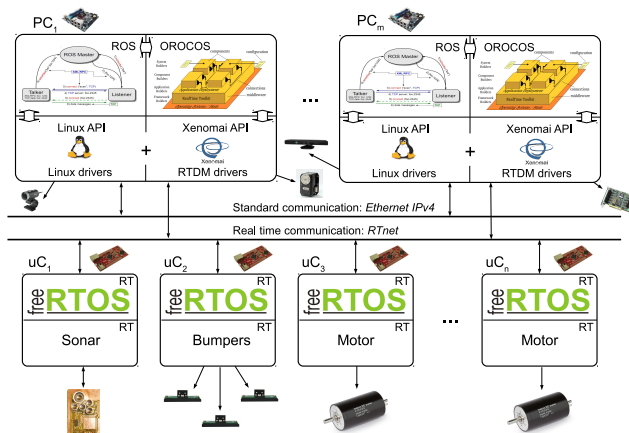
Scentralizowany





Wprowadzenie do systemów rozproszonych

Rozproszony





Wprowadzenie do systemów rozproszonych

Distributed System

- Zbiór niezależnych komputerów, który w działaniu postrzegany jest jako pojedynczy system
- Lamport's Definition
"System, w którym awaria komputera, o którym nigdy nie słyszałeś, powoduje że nie możesz wykonać roboty"
- Zbiór autonomicznych komputerów, dostarczających usług w zakresie przetwarzania informacji, zaspokajających potrzeby współczesnych przedsiębiorstw



Wprowadzenie do systemów rozproszonych

System rozproszony (cd.)

- Wiele autonomicznych komputerów
 - każdy wyposażony w CPU, lokalną pamięć, przestrzeń dyskową, infrastrukturę we/wy oddziałującą na otoczenie
 - rozproszone geograficznie
- Sieć połączeń
 - wybrane we/wy służą wymianie danych z innymi komputerami
- Dzielony stan
 - pamięć dzielona nie jest konieczna
 - system kooperuje w celu zarządzania wspólnym stanem
 - utrzymanie globalnych niezmienników wymaga prawidłowego i skoordynowanego działania wielu komputerów



Wprowadzenie do systemów rozproszonych

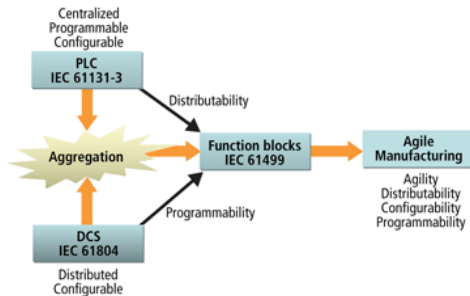
Komunikacja – wybrane problemy

- Szybki, niezawodny, fizyczny interfejs komunikacyjny
- Ograniczenia czasu rzeczywistego (wymagany RTOS) – software vs hardware
- Protokół komunikacji wielu-do-wielu – typowo zorientowany na komunikaty, model publikuj/subskrybuj.
- Quality of Service (QoS)
- Reprezentacja danych (*serialization*)
- Przenośność, dostępność (różne architektury, ogólnego zastosowania czy wbudowane , OS vs bare-metal)



Wprowadzenie do systemów rozproszonych

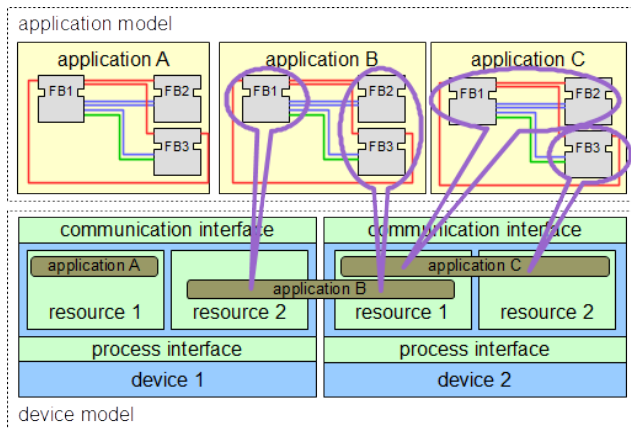
Norma IEC 61499 – historia



2

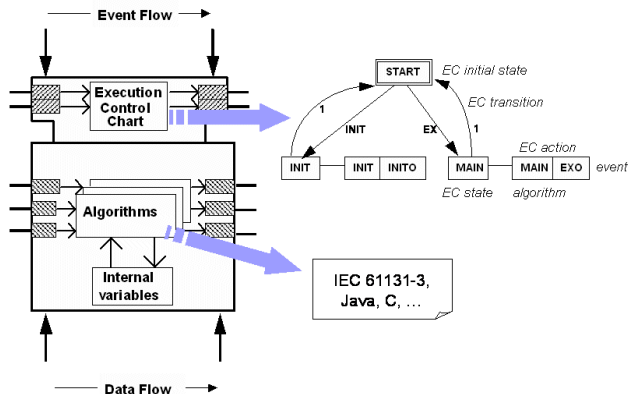
Wprowadzenie do systemów rozproszonych

Norma IEC 61499 – architektura



Wprowadzenie do systemów rozproszonych

Norma IEC 61499 – komponent





Wprowadzenie do systemów rozproszonych

BRICS: Best Practices in Robotics. *Podstawowym celem BRICS jest ustrukturyzowanie i sformalizowanie samego procesu rozwoju robota oraz dostarczanie narzędzi, modeli i bibliotek funkcjonalnych, które znacznie przyspieszają ten proces.*⁵

- Projekt europejski, którego celem było wypracowanie „najlepszych praktyk” dedykowanych wytwarzaniu systemów robotycznych
- Zbadania słabości dotychczasowych projektów robotycznych
- Badanie możliwości integracji sprzętu i oprogramowania
- Opracowanie IDE dla projektów robotycznych – BRIDE
- Definicja scenariuszy umożliwiających ewaluację opracowanych metodologii

⁵www.best-of-robotics.org



Wprowadzenie do systemów rozproszonych

Komponent

- podstawowa jednostka oprogramowania zdolna do samoistnego działania,
- jasno określone interfejsy i zależności,
- zapewnia dostęp do ustalonych funkcjonalności,
- może być parametryzowany bez konieczności zmiany kodu programu.

Podejście zorientowane na komponenty

- budowa oprogramowania z prefabrykowanych elementów,
- modułowość,
- elementy wielokrotnego użytku,
- łatwe utrzymanie i dostosowanie do nowych potrzeb.



Wprowadzenie do systemów rozproszonych

Oprogramowanie pośredniczące (*middleware*)⁶

- Uproszcza proces rozwoju oprogramowania
- Dostarcza mechanizmy komunikacji i współdziałania dla komponentów
- Zapewnia efektywne wykorzystanie dostępnych zasobów
- Wprowadza abstrakcję (sprzętu, języków programowania, OS)
- Umożliwia integrację z innymi systemami
- Dostarcza usługi bazowe
- Zapewnia automatyczne wykrywanie i konfigurację zasobów
- Wspiera komponenty wbudowane i niskopoziomowe urządzenia

⁶N. Mohamed, J. Al-Jaroodi, I. Jawhar, "Middleware for Robotics: A Survey," in Robotics, Automation and Mechatronics, 2008 IEEE Conference on , vol., no., pp.736-742, 21-24 Sept. 2008



Wprowadzenie do systemów rozproszonych

Robotyczne *middleware*

- OROCOS
- ROCK
- ORCA
- YARP
- OpenRTM
- OpenRave
- ROS
- Player
- ...



Wprowadzenie do systemów rozproszonych

- *Open Robot Control Software*
- Licencja otwartego oprogramowania
- Wspierane języki: C++, LUA i skryptowy
- Model programowy: COP (*Component Oriented Programming*)
- Elementy składowe
 - *Orocos Toolchain*
 - *Kinematics & Dynamics Library*
 - *Bayesian Filtering Library*
- Umożliwia komunikację i sterowanie w reżimie czasu rzeczywistego
- Integracja z Xenomai i ROS



Wprowadzenie do systemów rozproszonych

- *Robot Operating System*
- Licencja otwartego oprogramowania
- Wspierane języki: C++, Python
- Model programowy: zorientowany na agenty
- Komunikacja
 - *Topics* – model publish/subscribe
 - *Service* – zdalne wywołanie procedury
- Globalny serwer nazw i parametrów
- Dostępne komponenty programowe: usługa śledzenia układów współrzędnych, narzędzia do wizualizacji i debugowania, stos nawigacji i planowanie ruchu manipulatorów, ...



Wprowadzenie do systemów rozproszonych

Biblioteki programistyczne

- Algebra – LAPACK, Eigen, uBlas, GSL, MPL, MKL, ACML, ...
- ODE – Sundials, Odeint, GSL, Acado Toolkit, ...
- Optymalizacja – IPOPT, Acado Toolkit, CasADi, ...
- Automatyczne różniczkowanie – ADOL-C, CasADi, Acado Toolkit, ad, ...



Wprowadzenie do systemów rozproszonych

Biblioteki programistyczne (cd.)

- Planowanie ruchu – MoveIt, MPK, OMPL, OOPSMP, MSL, ...
- Wizja – OpenCV, PCL, LIBELAS, LIBVISO, VTK, ...
- Nawigacja – ROS Navigation Stack, MRPT, ...
- SLAM – MRPT, RobotVision, gmapping, ...
- Symulacja – Bullet Physics, ODE, AMD Havok, NVIDIA Physicks, ...
- Sterowanie – ros_control, OpenRTDynamics, Robotics Toolbox, ROCK ...
- ...



The End

Dziękuję za uwagę.