COM2009-3009 **Robotics**

Lecture 2

Robot Programming

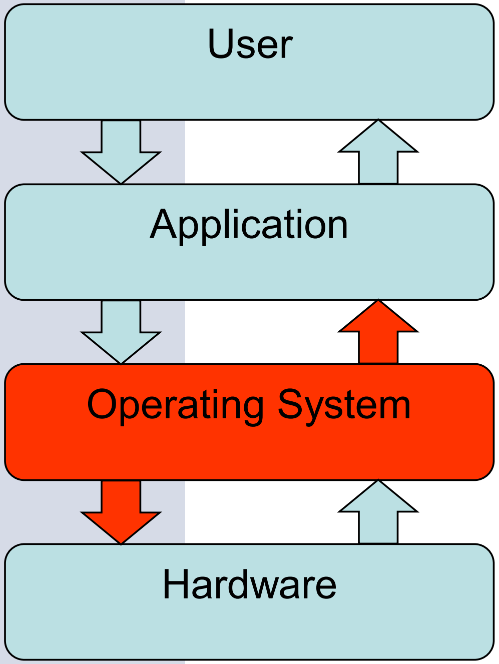
Dr Tom Howard

Multidisciplinary Engineering Education (MEE)

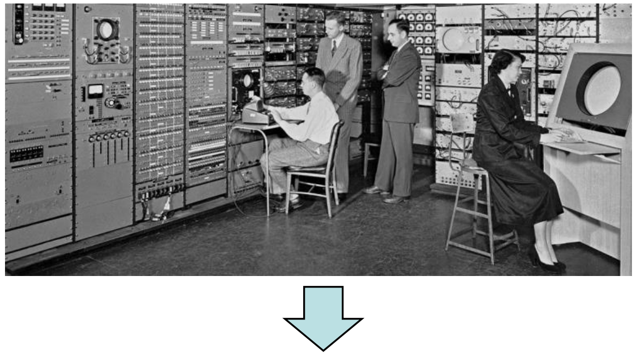
This lecture will cover …

1. Need for a generalised robot programming platform
2. History of the Robot Operating System (ROS)
3. ROS Basics
4. ROS Demo

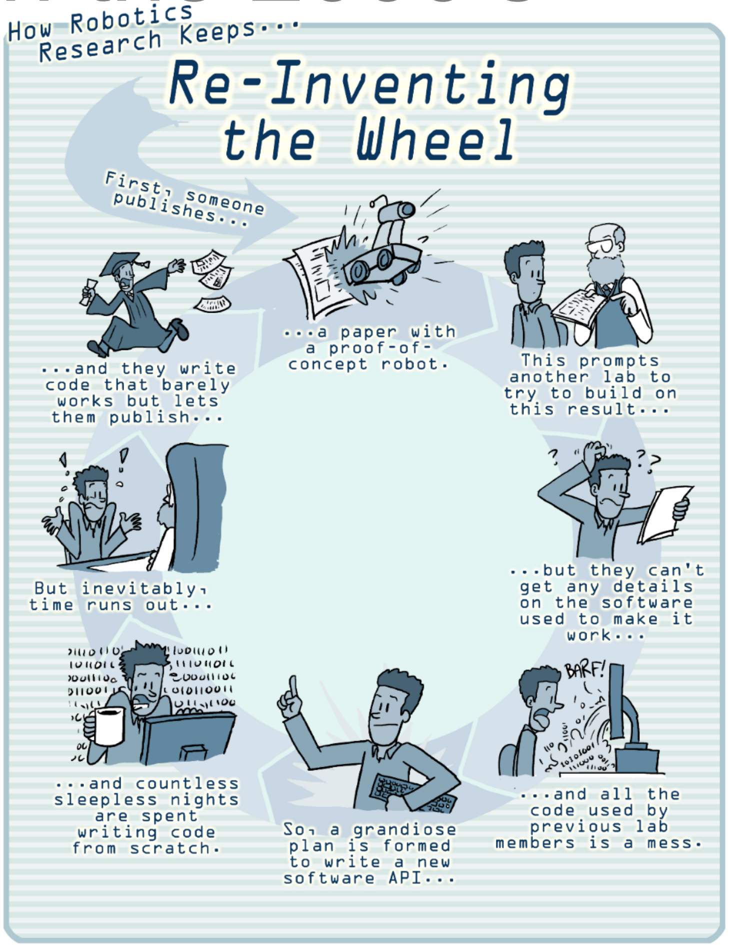
Why do OS exist?



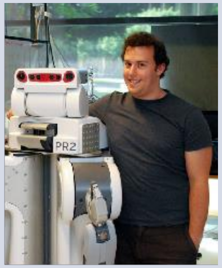
OS provide standardized interfaces facilitating hardware agnostic software.



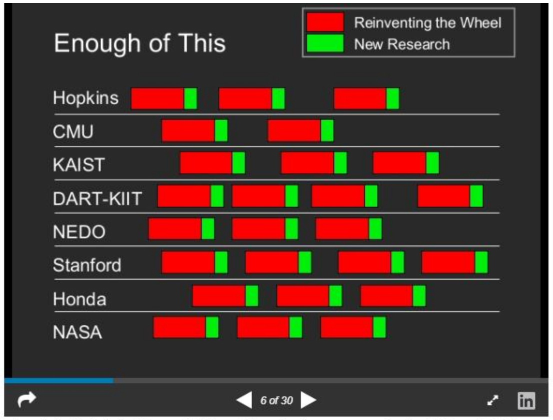
Robotics in the 2000's


How Robotics Research Keeps...
Re-Inventing the Wheel
First, someone publishes...
...a paper with a proof-of-concept robot.
This prompts another lab to try to build on this result...
...but they can't get any details on the software used to make it work...
BARF!
...and all the code used by previous lab members is a mess.
So, a grandiose plan is formed to write a new software API...
...and countless sleepless nights are spent writing code from scratch.
But inevitably, time runs out...
...and they write code that barely works but lets them publish...

Birth of the Robot Operating System (ROS)


Eric Berger

Keenan Wyrobek



ROS: *"the Linux of Robotics"*

www.redhat.com/en/open-source-stories/robots/breaking-the-wheel

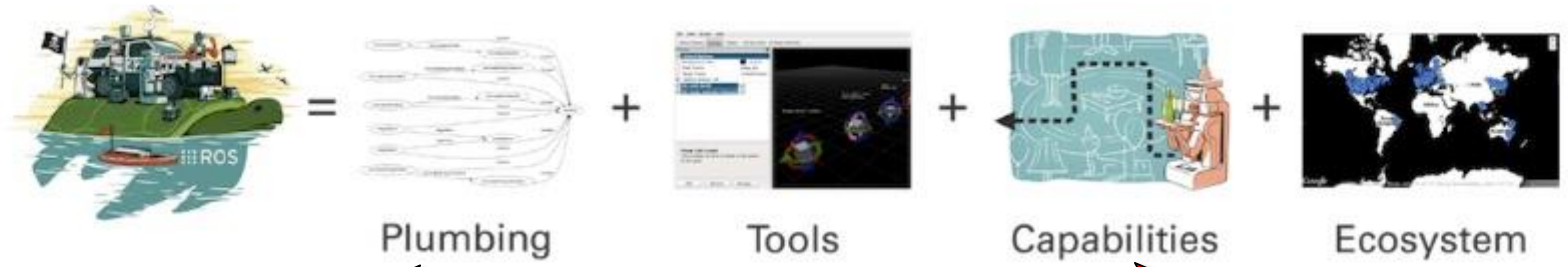
Designed and built 10* PR-1 robots then distributed to Univs to start dev community

What is ROS?

- Open source pseudo operating system (sits on top of standard OS)
- Collection of design/development tools for programming robots (Simulation & visualisation tools)
- Distributed architecture – not just communication between processes but also machines (e.g. PC to robot)
- Data Handling and Analysis
- Language independent – C++, Python (& Java, lisp, MATLAB, etc…)
- Implementation of a standard tools, and interfaces to different problems with the intention of re-usability

- Not an operating system, a programming language or an IDE.

ROS Overview



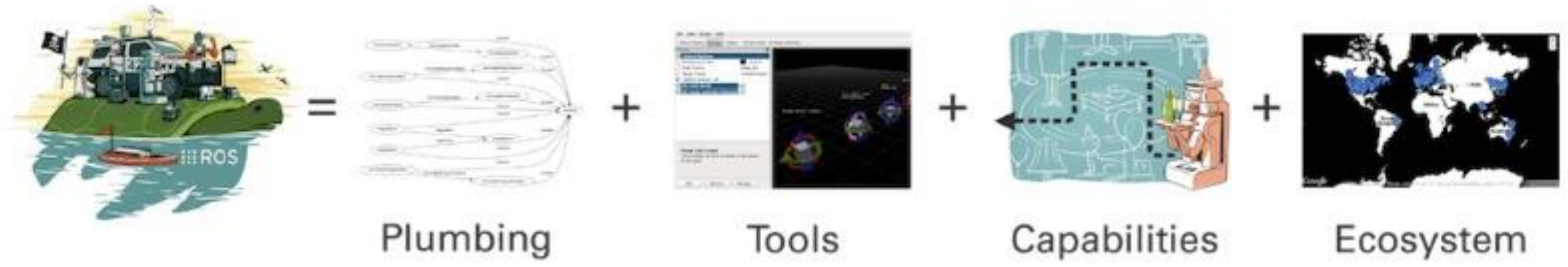
Key ROS functionality.

provides publish-subscribe messaging infrastructure designed to support the quick and easy construction of distributed computing systems.

Tools for configuring, starting, introspecting, debugging, visualizing, logging, testing & stopping distributed computing systems.

Verified implementations of useful robot functions, focused on manipulation, mobility, & perception.

ROS Overview



"…. according to ABI Research, roughly 55% of the world's robots will include a ROS package by 2024 [1]."

[1] https://www.abiresearch.com/market-research/product/1029218-open-source-robotics-projects/

ROS Basic Graph Structure

ROS Master

Handles distributed communications connections (inter thread / computer). Registers and looks up ROS resources.

ROS Node

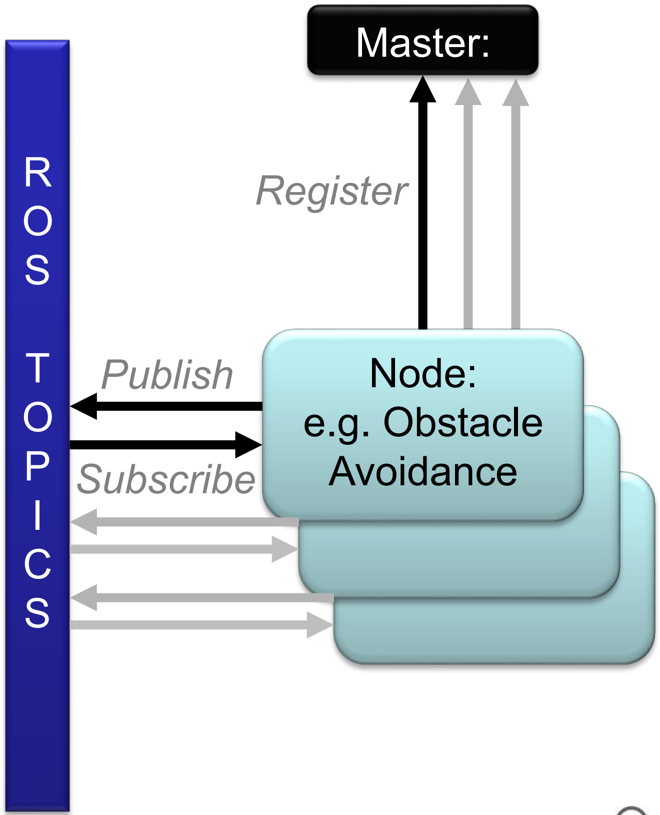
A process that performs computation.

e.g. Obstacle avoidance node: adapts motor speed proportionally to Laser Displacement Data

ROS Topic

Named bus over which nodes exchange messages. Asynchronous. Publish/Subscribe model.

e.g. subscribes to ***`/laser`*** *topic and publishes to* ***`/velocity`*** *topic.*

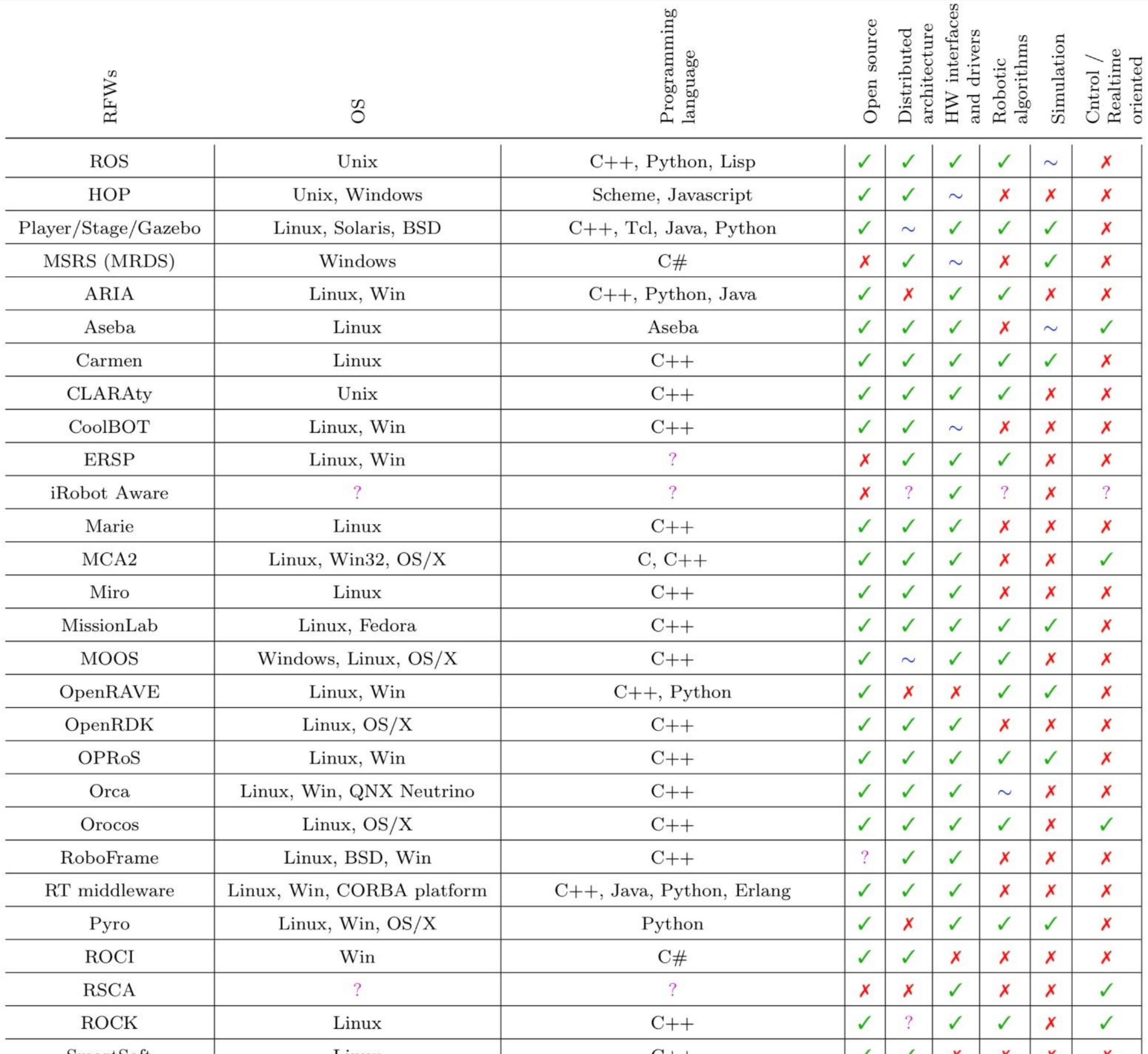


ROS 1 limitations

- **ROS development tied to OS development (Ubuntu)**
- **Real-time performance not guaranteed**
- **Not suited to multiple robots due to need for ROS Master**
- **Assumes good network connectivity & no security**
- **Can be too bloated for small robot applications**

But **ROS 2** promises to address these concerns.
`http://design.ros2.org/articles/why_ros2.html`

Other middlewares are available



RFWs	OS	Programming language	Open source	Distributed architecture	HW interfaces and drivers	Robotic algorithms	Simulation	Cntrol / Realtime oriented
ROS	Unix	C++, Python, Lisp	✓	✓	✓	✓	~	✗
HOP	Unix, Windows	Scheme, Javascript	✓	✓	~	✗	✗	✗
Player/Stage/Gazebo	Linux, Solaris, BSD	C++, Tcl, Java, Python	✓	~	✓	✓	✓	✗
MSRS (MRDS)	Windows	C#	✗	✓	~	✗	✓	✗
ARIA	Linux, Win	C++, Python, Java	✓	✗	✓	✓	✗	✗
Aseba	Linux	Aseba	✓	✓	✓	✗	~	✓
Carmen	Linux	C++	✓	✓	✓	✓	✓	✗
CLARAty	Unix	C++	✓	✓	✓	✓	✗	✗
CoolBOT	Linux, Win	C++	✓	✓	~	✗	✗	✗
ERSP	Linux, Win	?	✗	✓	✓	✓	✗	✗
iRobot Aware	?	?	✗	?	✓	?	✗	?
Marie	Linux	C++	✓	✓	✓	✗	✗	✗
MCA2	Linux, Win32, OS/X	C, C++	✓	✓	✓	✗	✗	✓
Miro	Linux	C++	✓	✓	✓	✗	✗	✗
MissionLab	Linux, Fedora	C++	✓	✓	✓	✓	✓	✗
MOOS	Windows, Linux, OS/X	C++	✓	~	✓	✓	✗	✗
OpenRAVE	Linux, Win	C++, Python	✓	✗	✗	✓	✓	✗
OpenRDK	Linux, OS/X	C++	✓	✓	✓	✗	✗	✗
OPRoS	Linux, Win	C++	✓	✓	✓	✓	✓	✗
Orca	Linux, Win, QNX Neutrino	C++	✓	✓	✓	~	✗	✗
Orocos	Linux, OS/X	C++	✓	✓	✓	✓	✗	✓
RoboFrame	Linux, BSD, Win	C++	?	✓	✓	✗	✗	✗
RT middleware	Linux, Win, CORBA platform	C++, Java, Python, Erlang	✓	✓	✓	✗	✗	✗
Pyro	Linux, Win, OS/X	Python	✓	✗	✓	✓	✓	✗
ROCI	Win	C#	✓	✓	✗	✗	✗	✗
RSCA	?	?	✗	✗	✓	✗	✗	✓
ROCK	Linux	C++	✓	?	✓	✓	✗	✓

Tsardoulias, E., Mitkas, A.P. (2017). Robotic frameworks, architectures and middleware comparison, `https://arxiv.org/abs/1711.06842`

Meet your robot