Principles of Software Construction:
Objects, Design, and Concurrency

Toward software engineering in practice

Charlie Garrod **Chris Timperley**

Carnegie Mellon University
School of Computer Science

institute for SOFTWARE RESEARCH

Administrivia

- Homework 5c due tonight!

Software Engineering (SE) at CMU

- 17-214: Code-level design
  - Extensibility, reuse, concurrency, functional correctness
- 17-313: Human aspects of software development
  - Requirements, teamwork, scalability, security, scheduling, costs, risks, business models
- 17-413 Practicum, 17-415 Seminar, Internship
- Various courses on requirements, architecture, software analysis, SE for startups, etc.
- SE Minor: http://isri.cmu.edu/education/undergrad

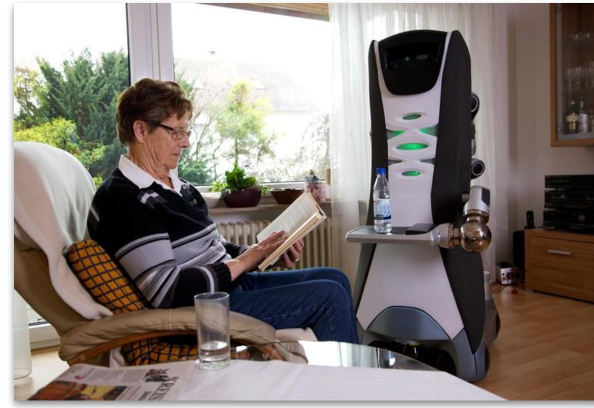
Major topics in 17-313 (Foundations of SE)

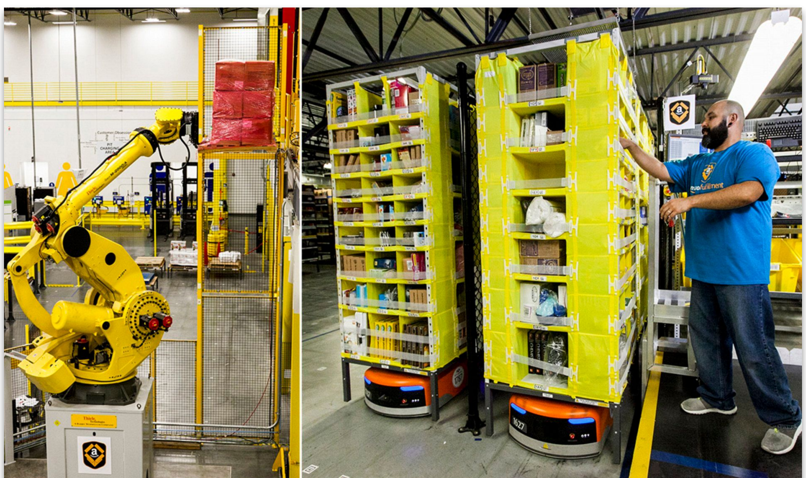
- Process considerations for software development
- Requirements elicitation, documentation, and evaluation
- Design for quality attributes
- Strategies for quality assurance
- Empirical methods in software engineering
- Time and team management
- Software engineering meets machine learning
- Economics of software development

Today: Software engineering in practice

- Software engineering for robotics
- Software testing for robotics
- Robot Operating System

Robotic systems are an increasingly important part of our lives





https://images.axios.com/aonz5kKRm3gZBcbX6qFI0Ma8r4k=/0x0:3000x1688/1920x1080/2018/12/03/1543836353174.jpg
https://i.dailymail.co.uk/i/newpix/2018/06/05/15/4CF0488F00000578-0-image-a-30_1528207334623.jpg
https://cdn.vox-cdn.com/thumbor/rkkbG6zOPUsjPysihu1iOiar2S0=/0x0:1016x677/1400x1050/filters:focal(427x258:589x420):format(jpeg)/cdn.vox-cdn.com/uploads/chorus_image/image/56360029/blake_dowling_3.0.jpg
http://www.newelectronics.co.uk/article-images/199663/Care-O-bot%20_popup.jpg
https://cbsnews1.cbsistatic.com/hub/i/2016/05/19/3b64ecc7-3da0-453f-a745-a22f26f7b27c/uatc-car-bridge-16x9-917x516.jpg

institute for SOFTWARE RESEARCH

How a Self-Driving Uber Killed a Pedestrian in Arizona

By TROY GRIGGS and DAISUKE WAKABAYASHI UPDATED MARCH 21, 2018

A woman was struck and killed on Sunday night by an autonomous car operated by Uber in Tempe, Ariz. It was believed to be the first pedestrian death associated with self-driving technology.

What We Know About the Accident

BBC | Sign in | News | Sport | Reel | Worklife | Travel | Future | Mo

NEWS

Home | Video | World | US & Canada | UK | Business | Tech | Science | Stories | Entertai

Business | Market Data | Global Trade | Companies | Entrepreneurship | Technology of Busir

Uber in fatal crash had safety flaws say US investigators

6 November 2019

Share

An Uber self-driving test vehicle that hit and killed a woman in 2018 had software problems, according to US safety investigators.

Elaine Herzberg, 49, was hit by the car as she was crossing a road in Tempe, Arizona.

The US National Transportation Safety Board (NTSB) found the car failed to identify her properly as a pedestrian.

The detailed findings raised a series of safety issues but did not determine the probable cause of the accident.

https://www.nytimes.com/interactive/2018/03/20/us/self-driving-uber-pedestrian-killed.html?mtrref=www.google.com&assetType=REGIWALL
https://www.bbc.com/news/business-50312340
https://www.bbc.com/news/technology-44243118

Search **Bloomberg** Sig

Technology

Boeing's 737 Max Software Outsourced to $9-an-Hour Engineers

By Peter Robison
June 28, 2019, 4:46 PM EDT

- Planemaker and suppliers used lower-paid temporary workers
- Engineers feared the practice meant code wasn't done right

The cockpit of a grounded 737 Max 8 aircraft. *Photographer: Dimas*

It remains the mystery at the hea
crisis: how a company renowned
made seemingly basic software m
deadly crashes. Longtime Boeing
was complicated by a push to ou
contractors.

The Max software -- plagued by is
planes grounded months longer
week revealed a new flaw -- was d
was laying off experienced engin
suppliers to cut costs.

https://spectrum.ieee.org/aerospace/aviation/how-the-boeing-7

A year after the first 737 Max crash, it's unclear when the plane will fly again

Two crashes of Boeing's 737 Max 8 killed 346 people, and authorities are blaming Boeing's design, a faulty sensor and airline staff. Plus: Everything you need to know about the plane.

Kent German November 1, 2019 9:01 AM PDT 63

ed killing 346 people.

ts 737 Max 8 that killed 346 people, Boeing is facing
its newest and most critical aircraft models. The
und the world, and the Federal Aviation

How the Boeing 737 Max Disaster Looks to a Software Developer

Design shortcuts meant to make a new plane seem like an old, familiar one are to blame

By **Gregory Travis**

The views expressed here are solely those of the author and do not represent positions of IEEE Spectrum *or the IEEE.*

Photo: Jemal Countess/Getty Images

This is part of the wreckage of Ethiopian Airlines Flight ET302, a Boeing 737 Max

How would you develop software for a delivery robot?

- What are the requirements of your system?
- Who are your stakeholders?
- **What software components might you need?**
- How do you safely glue together those components?
- What assumptions are you making?

Sidewalk delivery robots coming to Pitt this fall

TRIB LIVE MATT MAIELLI | Monday, September 30, 2019 12:01 a.m.

PHOTO COURTESY OF STARSHIP TECHNOLOGIES

Starship Technologies and University of Pittsburgh plan to deploy a fleet of autonomous delivery robots in Oakland to deliver to students.

The sidewalks in Oakland may soon be getting even more crowded.

The University of Pittsburgh and Starship Technologies, a robotics company that makes 50-pound robots resembling coolers on wheels, plan to launch a delivery service this year.

"We are working with the city and our Oakland neighbors toward a launch later this fall," Pitt spokesman Kevin Zwick said in an email.

A fleet of about 25 robots will be programmed to travel on Pitt's campus carrying groceries, take-out meals and packages.

The Oakland Planning and Development Corporation discussed bringing automated delivery robots to the neighborhood during an August meeting. David Catania, Starship's head of government affairs, reported then that the robots would operate in a similar manner to other universities.

Robots will deliver food and drinks to George Mason University students

The college taps ground drone startup Starship Technologies to make snack runs

By Sean O'Kane | @sokane1 | Jan 22, 2019, 5:07pm EST

SHARE

Photo: Sean O'Kane / The Verge

European startup Starship Technologies is bringing its six-wheeled delivery robots to a college campus in Virginia. The company announced Monday that George Mason University will allow students to use their meal plans to have select food and drink orders delivered by the robots.

Starship says it's providing George Mason with at least 25 robots, and orders from Blaze Pizza, Starbucks, and Dunkin' will be available at the start. More will be added in the "coming weeks," and each order will cost $1.99 extra.

Amazon delivery robots are officially on the streets of California

Amazon has robots on the streets. It's a good bet urban delivery will never be the same.

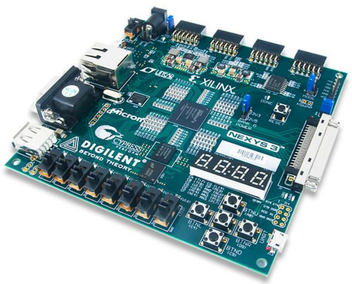
By Greg Nichols for Robotics | August 7, 2019 -- 11:00 GMT (04:00 PDT) | Topic: Robotics

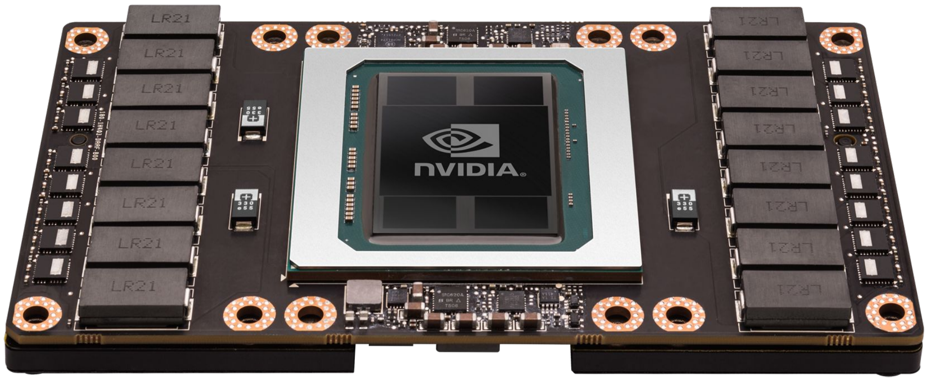
Amazon has officially rolled out its last-mile delivery robots in a Southern California testbed. Called Scout, the delivery robot is designed to autonomously ferry parcels from urban distribution points to Amazon Prime customers, removing the need for vans and cars in last-mile delivery.

SEE: Autonomous vehicles and the enterprise **(ZDNet/TechRepublic special feature) |** Download the free PDF version **(TechRepublic)**

https://triblive.com/local/pittsburgh-allegheny/sidewalk-delivery-robots-coming-to-pitt-this-fall/
https://www.theverge.com/2019/1/22/18193391/robots-delivery-george-mason-university-students-dunkin-starbucks-blaze-pizza
https://www.zdnet.com/article/amazon-delivery-robots-are-officially-on-the-streets-of-california/

Robotics software engineering is all about integration





https://upload.wikimedia.org/wikipedia/commons/thumb/c/c3/Python-logo-notext.svg/1024px-Python-logo-notext.svg.png
https://raw.githubusercontent.com/isocpp/logos/master/cpp_logo.png
https://upload.wikimedia.org/wikipedia/commons/f/f4/Lisplogo.png
https://cdn10.bigcommerce.com/s-7gavg/products/104/images/5166/Nexys3-obl_2-600__12570.1536184396.1280.1280.jpg?c=2
https://www.engadget.com/2014/12/31/oroginal-kinect-discontinued/

Metrics of software quality, i.e., *design goals*

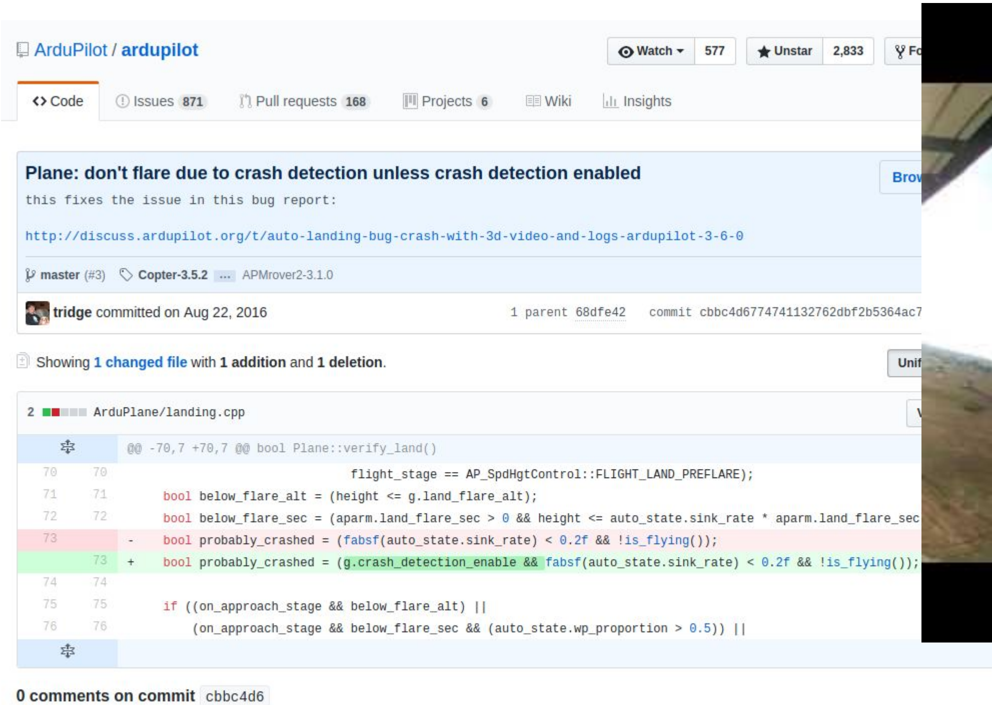
Functional correctness	Adherence of implementation to the specifications
Robustness	Ability to handle anomalous events
Flexibility	Ability to accommodate changes in specifications
Reusability	Ability to be reused in another application
Efficiency	Satisfaction of speed and storage requirements
Scalability	Ability to serve as the basis of a larger version of the application
Security	Level of consideration of application security

Source: Braude, Bernstein, Software Engineering. Wiley 2011

Today: Software engineering in practice

- Software engineering for robotics
- Software testing for robotics
- Robot Operating System

Could we have prevented this bug?



https://www.youtube.com/watch?v=Rjjj6DAylsk

institute for SOFTWARE RESEARCH

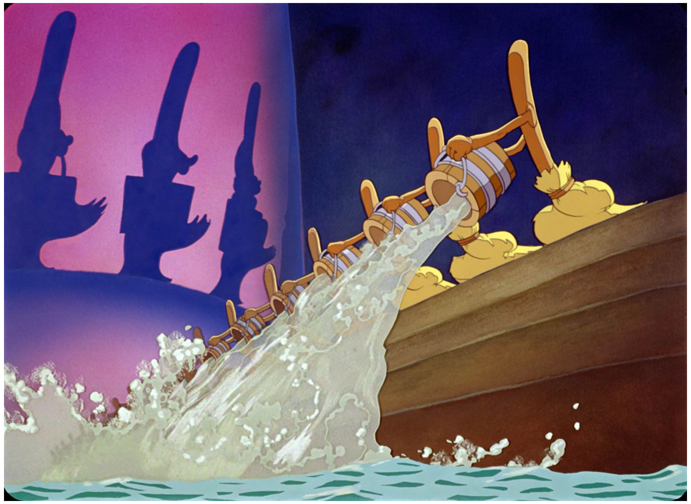
Unit Testing

When is unit testing not enough?

How should we test our robotics software?

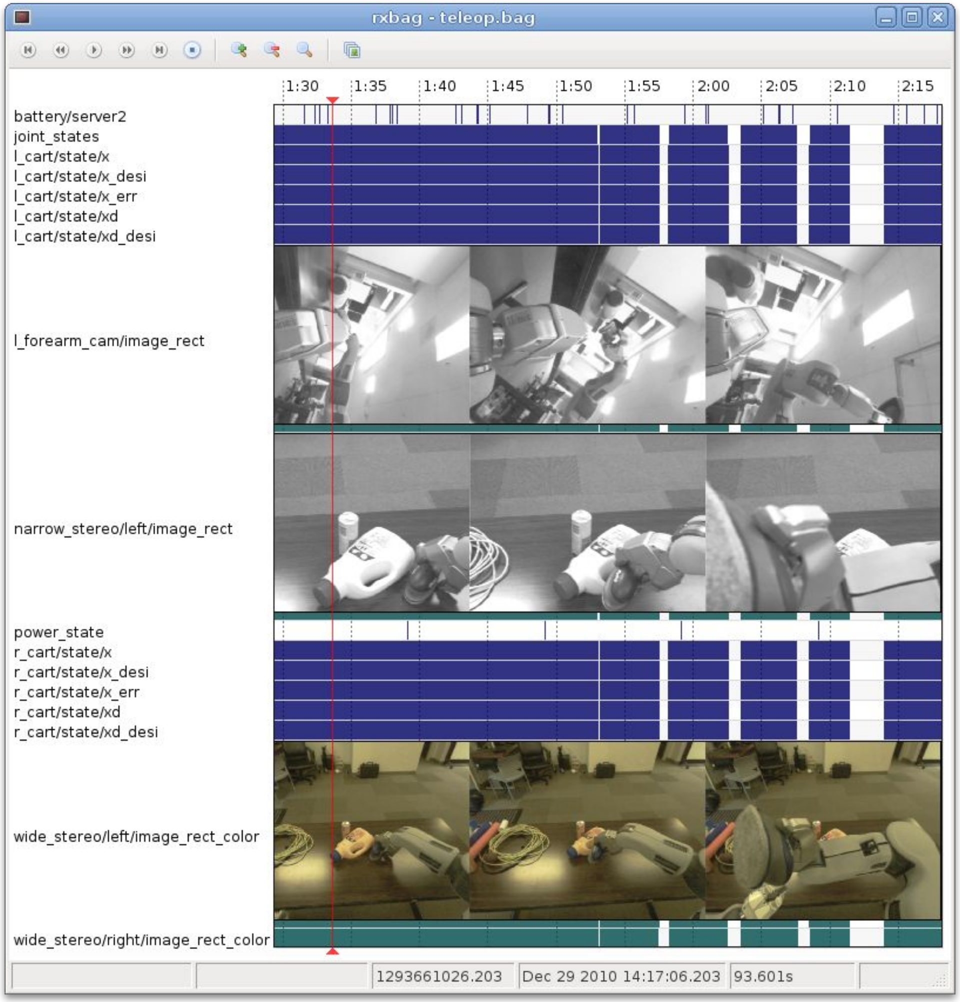
Challenges for testing robotics

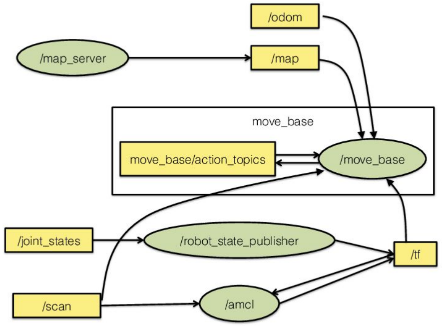
- It's really expensive! Requires substantial time and resources
- It's dangerous!
- Test setup complexity
- Unpredictable corner cases
- **The "oracle problem" is even harder**
  - How do we know that the robot did the right thing?
  - How do we know that the robot didn't do a bad thing?
- **Cultural and economic issues**
  - Lack of incentives
  - Emphasis on results rather than quality
- ...



http://www.thearttheater.org/wp-content/uploads/2015/10/Fantasia-brooms.jpg
https://www.youtube.com/watch?v=3hKgEylk8ks

Record-and-Replay Testing

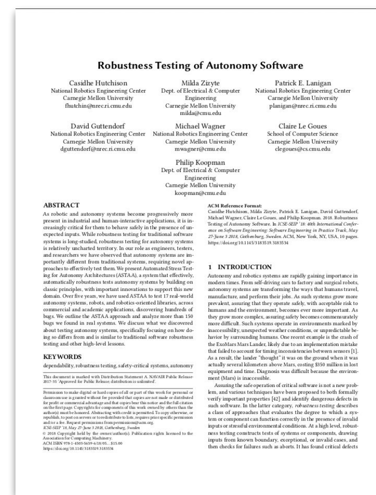




https://wiki.ros.org/rxbag
https://www.ros.org/news/2010/03/whats-in-the-box-logging-and-playback-with-rosbag.html
https://20kh6h3g46l33ivuea3rxuyu-wpengine.netdna-ssl.com/wp-content/uploads/2019/08/1I3NMNgKO9A8W0ww0mgWFBQ.png

Robustness Testing (a.k.a. Stress Testing)

Conceptually similar to *fuzzing*.



Robustness Testing of Autonomy Software

Casidhe Hutchison
National Robotics Engineering Center
Carnegie Mellon University
fhutchin@nrec.ri.cmu.edu

Milda Zizyte
Dept. of Electrical & Computer Engineering
Carnegie Mellon University
milda@cmu.edu

Patrick E. Lanigan
National Robotics Engineering Center
Carnegie Mellon University
planigan@nrec.ri.cmu.edu

David Guttendorf
National Robotics Engineering Center
Carnegie Mellon University
dguttendorf@nrec.ri.cmu.edu

Michael Wagner
National Robotics Engineering Center
Carnegie Mellon University
mwagner@cmu.edu

Claire Le Goues
School of Computer Science
Carnegie Mellon University
clegoues@cs.cmu.edu

Philip Koopman
Dept. of Electrical & Computer Engineering
Carnegie Mellon University
koopman@cmu.edu

ABSTRACT

As robotic and autonomy systems become progressively more present in industrial and human-interactive applications, it is increasingly critical for them to behave safely in the presence of unexpected inputs. While robustness testing for traditional software systems is long-studied, robustness testing for autonomy systems is relatively uncharted territory. In our role as engineers, testers, and researchers we have observed that autonomy systems are importantly different from traditional systems, requiring novel approaches to effectively test them. We present Automated Stress Testing for Autonomy Architectures (ASTAA), a system that effectively, automatically robustness tests autonomy systems by building on classic principles, with important innovations to support this new domain. Over five years, we have used ASTAA to test 17 real-world autonomy systems, robots, and robotics-oriented libraries, across commercial and academic applications, discovering hundreds of bugs. We outline the ASTAA approach and analyze more than 150 bugs we found in real systems. We discuss what we discovered about testing autonomy systems, specifically focusing on how doing so differs from and is similar to traditional software robustness testing and other high-level lessons.

KEYWORDS

dependability, robustness testing, safety-critical systems, autonomy

This document is marked with Distribution Statement A. NAVAIR Public Release 2017-35 'Approved for Public Release; distribution is unlimited'.

ICSE-SEIP '18, May 27-June 3 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5659-6/18/05...$15.00
https://doi.org/10.1145/3183519.3183534

ACM Reference Format:
Casidhe Hutchison, Milda Zizyte, Patrick E. Lanigan, David Guttendorf, Michael Wagner, Claire Le Goues, and Philip Koopman. 2018. Robustness Testing of Autonomy Software. In *ICSE-SEIP '18: 40th International Conference on Software Engineering: Software Engineering in Practice Track, May 27-June 3 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3183519.3183534

1 INTRODUCTION

Autonomy and robotics systems are rapidly gaining importance in modern times. From self-driving cars to factory and surgical robots, autonomy systems are transforming the ways that humans travel, manufacture, and perform their jobs. As such systems grow more prevalent, assuring that they operate safely, with acceptable risk to humans and the environment, becomes ever more important. As they grow more complex, assuring safety becomes commensurately more difficult. Such systems operate in environments marked by inaccessibility, unexpected weather conditions, or unpredictable behavior by surrounding humans. One recent example is the crash of the ExoMars Mars Lander, likely due to an implementation mistake that failed to account for timing inconsistencies between sensors [1]. As a result, the lander "thought" it was on the ground when it was actually several kilometers above Mars, costing $350 million in lost equipment and time. Diagnosis was difficult because the environment (Mars) is inaccessible.

Assuring the safe operation of critical software is not a new problem, and various techniques have been proposed to both formally verify important properties [42] and identify dangerous defects in such software. In the latter category, *robustness testing* describes a class of approaches that evaluates the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions. At a high level, robustness testing constructs tests of systems or components, drawing inputs from known boundary, exceptional, or invalid cases, and then checks for failures such as aborts. It has found critical defects

https://edge-case-research.com/
http://safeautonomy.blogspot.com/2018/07/robustness-testing-of-autonomy-software.html
https://www.nrec.ri.cmu.edu/

End-to-End Testing: Field Testing

https://i2.wp.com/agfax.com/wp-content/uploads/maxresdefault-e1505256365516.jpg?fit=640%2C360&ssl=1
https://cbsnews1.cbsistatic.com/hub/i/2016/05/19/3b64ecc7-3da0-453f-a745-a22f26f7b27c/uatc-car-bridge-16x9-917x516.jpg
https://www.cmu.edu/news/archive/2008/October/oct14_scarabhawaii.shtml
https://news.engin.umich.edu/2018/03/m-air-autonomous-aerial-vehicle-outdoor-lab-opens/

~$350M software bug reproduced in simulation


make a contribution subscribe find a job

theguardian

news / opinion / sport / arts / life

science / US / world / environment / US politics / business / more

Mars

How ExoMars Schiaparelli lander may have met its fate on Mars

European Space Agency not immediately certain that probe crashed but early data analysis indicates destructive impact on red planet

The data beamed back from Schiaparelli show that the first five minutes of its descent went flawlessly. Illustration: Reuters

593 521

This article is 1 year old

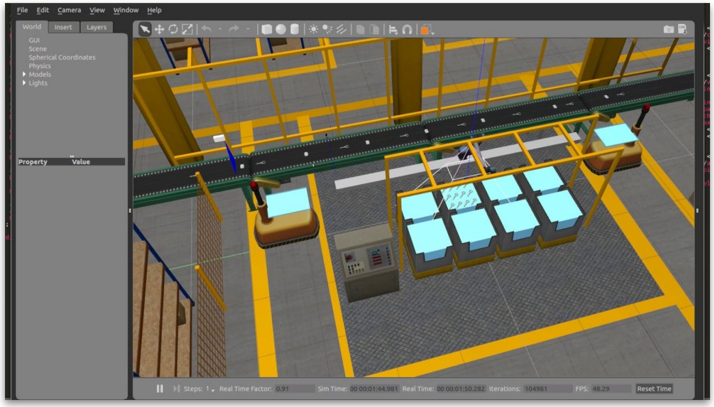
Hannah Devlin Science correspondent

Thursday 20 October 2016 10.41 EDT

It was supposed to be the first European spacecraft to carry out science on Mars,

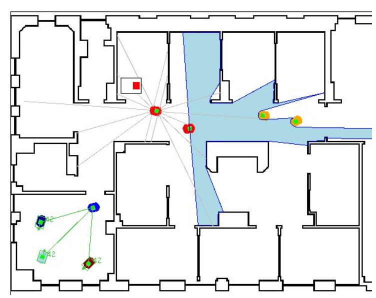
institute for SOFTWARE RESEARCH

Simulation-based testing: Software-in-the-loop



GAZEBO

UNREAL ENGINE

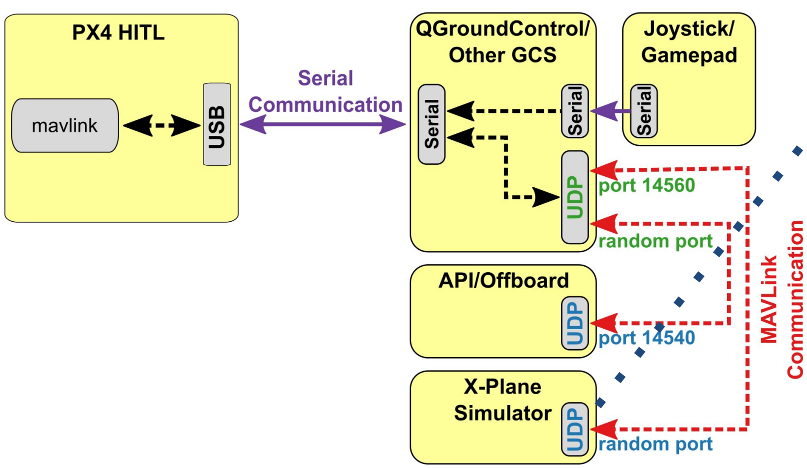


amazon web services

PlayerStage

https://i.ytimg.com/vi/FYi8grwE-zE/maxresdefault.jpg
https://storage.googleapis.com/groundai-web-prod/media%2Fusers%2Fuser_75855%2Fproject_61467%2Fimages%2Fdrone_depth_materials.png
http://playerstage.sourceforge.net/stage/stage.html

Simulation-based testing: Hardware-in-the-loop



https://www.researchgate.net/figure/Typical-Hardware-In-The-Loop-HITL-testbed-configuration-25_fig1_331103529
https://blog.jiashen.me/2015/06/30/uav-software-recipes-i-hardward-in-the-loop-simulation/
https://dev.px4.io/v1.9.0/en/simulation/hitl.html
http://vrthegamers.com/x-plane-11-20-patch-adds-native-vr-support/#.XdHZmNFOlhE
https://cdn.getfpv.com/media/catalog/product/cache/1/image/9df78eab33525d08d6e5fb8d27136e95/p/i/pixhawk4-main_1_1.jpg

Simulation-based testing: Hybrid

https://www.orbisprotect.com/empty-warehouse/

~50% of bugs can be detected with low-fidelity simulation

Only **14%** of bugs rely on the presence of physical hardware
(e.g., lights and sounds)

Only **10%** of bugs depend upon environmental factors
(e.g., human arm)

72% of bugs can be triggered using a single form of discrete input.

Only **5%** of bugs require concurrent events in order to be triggered

36% of bugs occur under a particular configuration

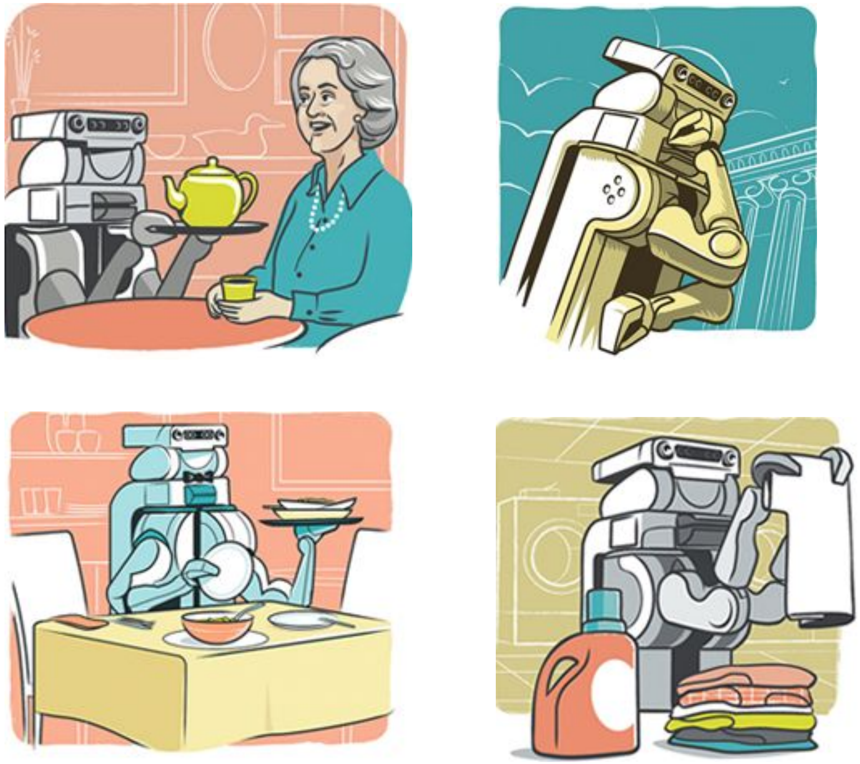
89% of bugs occur during normal operating conditions

Crashing Simulated Planes is Cheap: Can Simulation Detect Robotics Bugs Early?, Christopher Steven Timperley, Afsoon Afzal, Deborah Katz, Jam Marcos Hernandez, and Claire Le Goues, in International Conference on Software Testing, Validation and Verification, ICST '18, 2018, pp. 331–342.

Today: Software engineering in practice

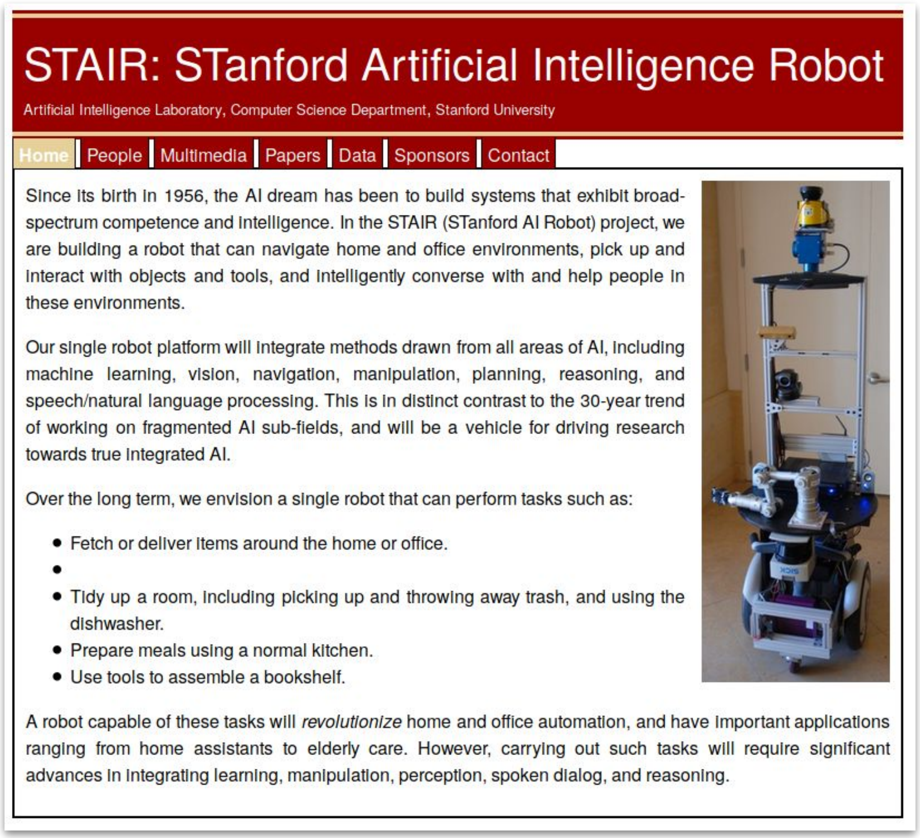
- Software engineering for robotics
- Software testing for robotics
- Robot Operating System

Eric Berger and Keenan Wyrobek were PhD students, working on building a platform for personal robotics



http://www.willowgarage.com/pages/pr2/applications

STAIR and the Stanford AI Lab



STAIR: STanford Artificial Intelligence Robot

Artificial Intelligence Laboratory, Computer Science Department, Stanford University

Home | People | Multimedia | Papers | Data | Sponsors | Contact

Since its birth in 1956, the AI dream has been to build systems that exhibit broad-spectrum competence and intelligence. In the STAIR (STanford AI Robot) project, we are building a robot that can navigate home and office environments, pick up and interact with objects and tools, and intelligently converse with and help people in these environments.

Our single robot platform will integrate methods drawn from all areas of AI, including machine learning, vision, navigation, manipulation, planning, reasoning, and speech/natural language processing. This is in distinct contrast to the 30-year trend of working on fragmented AI sub-fields, and will be a vehicle for driving research towards true integrated AI.

Over the long term, we envision a single robot that can perform tasks such as:

- Fetch or deliver items around the home or office.
-
- Tidy up a room, including picking up and throwing away trash, and using the dishwasher.
- Prepare meals using a normal kitchen.
- Use tools to assemble a bookshelf.

A robot capable of these tasks will *revolutionize* home and office automation, and have important applications ranging from home assistants to elderly care. However, carrying out such tasks will require significant advances in integrating learning, manipulation, perception, spoken dialog, and reasoning.

http://stair.stanford.edu

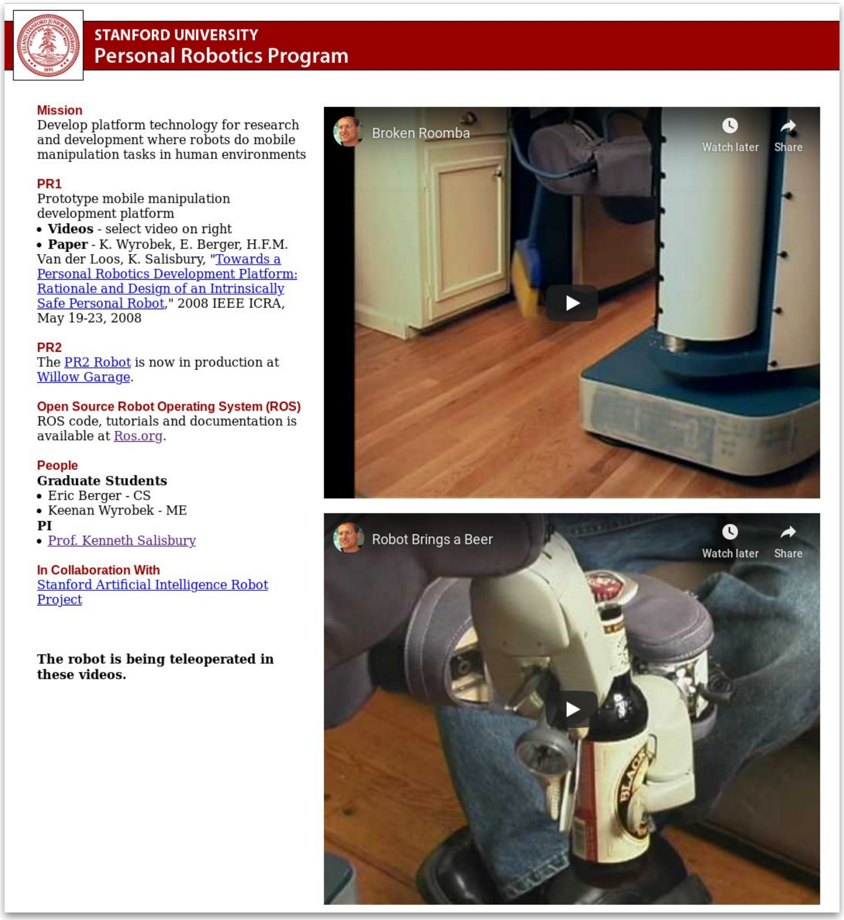
Robotics had a code reuse problem

A slide from the original pitch!

https://www.theconstructsim.com/history-ros/
https://spectrum.ieee.org/automaton/robotics/robotics-software/the-origin-story-of-ros-the-linux-of-robotics

Personal Robotics Program and PR1

PR1



STANFORD UNIVERSITY
Personal Robotics Program

Mission
Develop platform technology for research and development where robots do mobile manipulation tasks in human environments

PR1
Prototype mobile manipulation development platform
- **Videos** - select video on right
- **Paper** - K. Wyrobek, E. Berger, H.F.M. Van der Loos, K. Salisbury, "Towards a Personal Robotics Development Platform: Rationale and Design of an Intrinsically Safe Personal Robot," 2008 IEEE ICRA, May 19-23, 2008

PR2
The PR2 Robot is now in production at Willow Garage.

Open Source Robot Operating System (ROS)
ROS code, tutorials and documentation is available at Ros.org.

People
Graduate Students
- Eric Berger - CS
- Keenan Wyrobek - ME

PI
- Prof. Kenneth Salisbury

In Collaboration With
Stanford Artificial Intelligence Robot Project

The robot is being teleoperated in these videos.

Broken Roomba
Watch later
Share

Robot Brings a Beer
Watch later
Share

2006

Spinoff: PR2 and Willow Garage

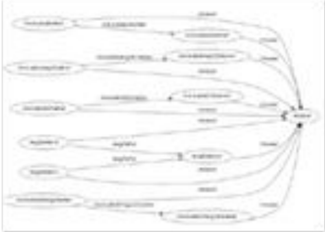
2007

https://robots.ieee.org/robots/pr2/Interactive%201/Media%20Player/SD-Q3-M360/pr2-int1-01.jpg
http://www.willowgarage.com/

“The Linux of Robotics”: Robot Operating System

Robot Operating System

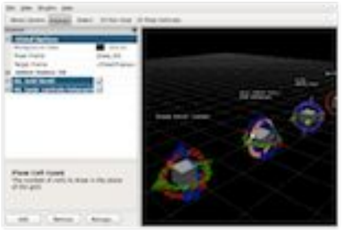
ROS



Plumbing

- Process management
- Inter-process communication
- Device drivers

+



Tools

- Simulation
- Visualization
- Graphical user interface
- Data logging

+

Capabilities

- Control
- Planning
- Perception
- Mapping
- Manipulation

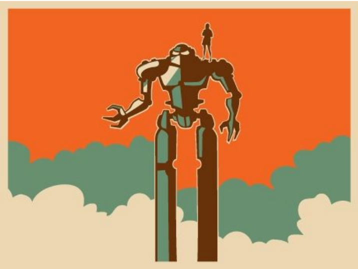
+

Ecosystem

- Package organisation
- Software distribution
- Documentation
- Tutorials

Exponential growth in the power of APIs

Without them, ROS wouldn't be possible!



'50s-'60s – Arithmetic.

'70s – malloc, bsearch, qsort, rnd, I/O, system calls, formatting, early databases

'80s – GUIs, desktop publishing, relational databases

'90s – Networking, multithreading, 3D graphics

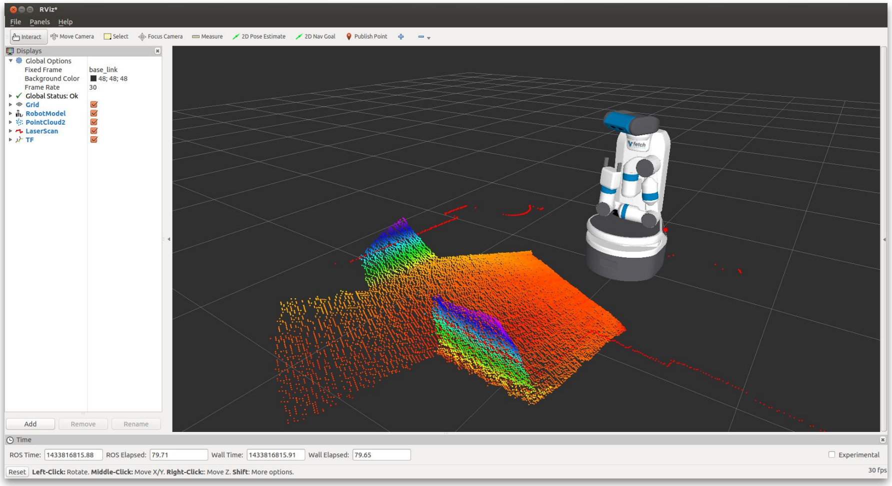
'00s – Data structures, higher-level abstractions, Web APIs: social media, cloud infrastructure

'10s – Machine learning, computer vision, IOT, robotics, pretty much everything

ROS is inherently collaborative

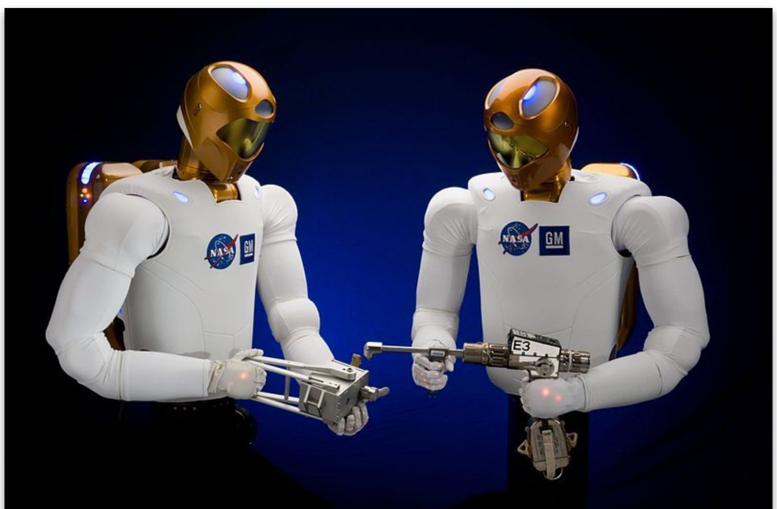
https://www.ros.org/wp-content/uploads/2013/12/user_map.jpg
https://i.ytimg.com/vi/qXZt-B7iUyw/maxresdefault.jpg

You don’t even need a robot!



https://fetchrobotics.com

ROS in production

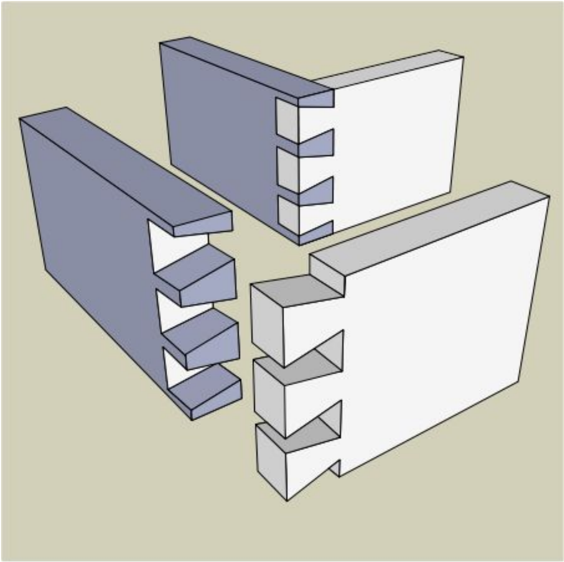


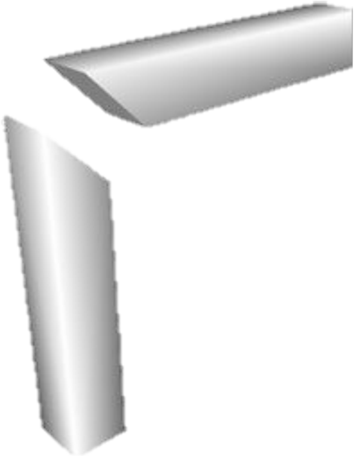
ROSin

https://en.wikipedia.org/wiki/Robonaut
https://assets.newatlas.com/dims4/default/3cbf0d2/2147483647/strip/true/crop/2048x1367+0+85/resize/1160x774!/quality/90/?url=https%3A%2F%2Fassets.newatlas.com%2Farchive%2Flaser-paint-remover.jpg
https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/ROS-Industrial-for-Real-World-Solutions/content_id/7919

Software architecture for ROS

Design Patterns





Architectural styles



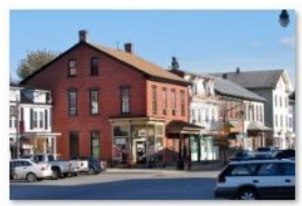










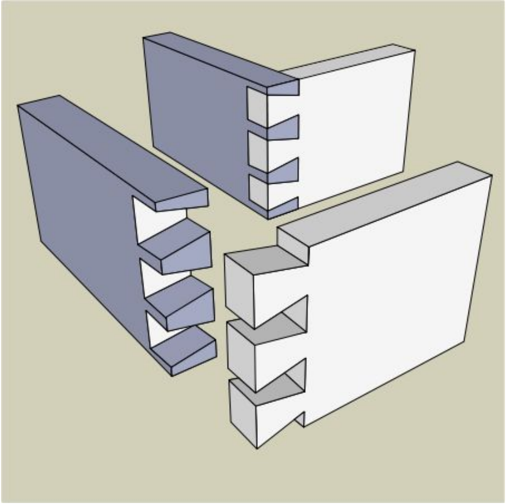


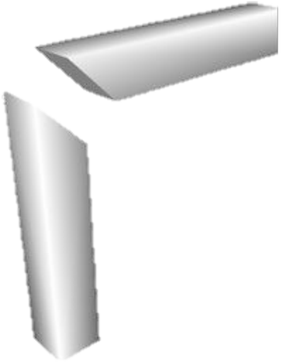




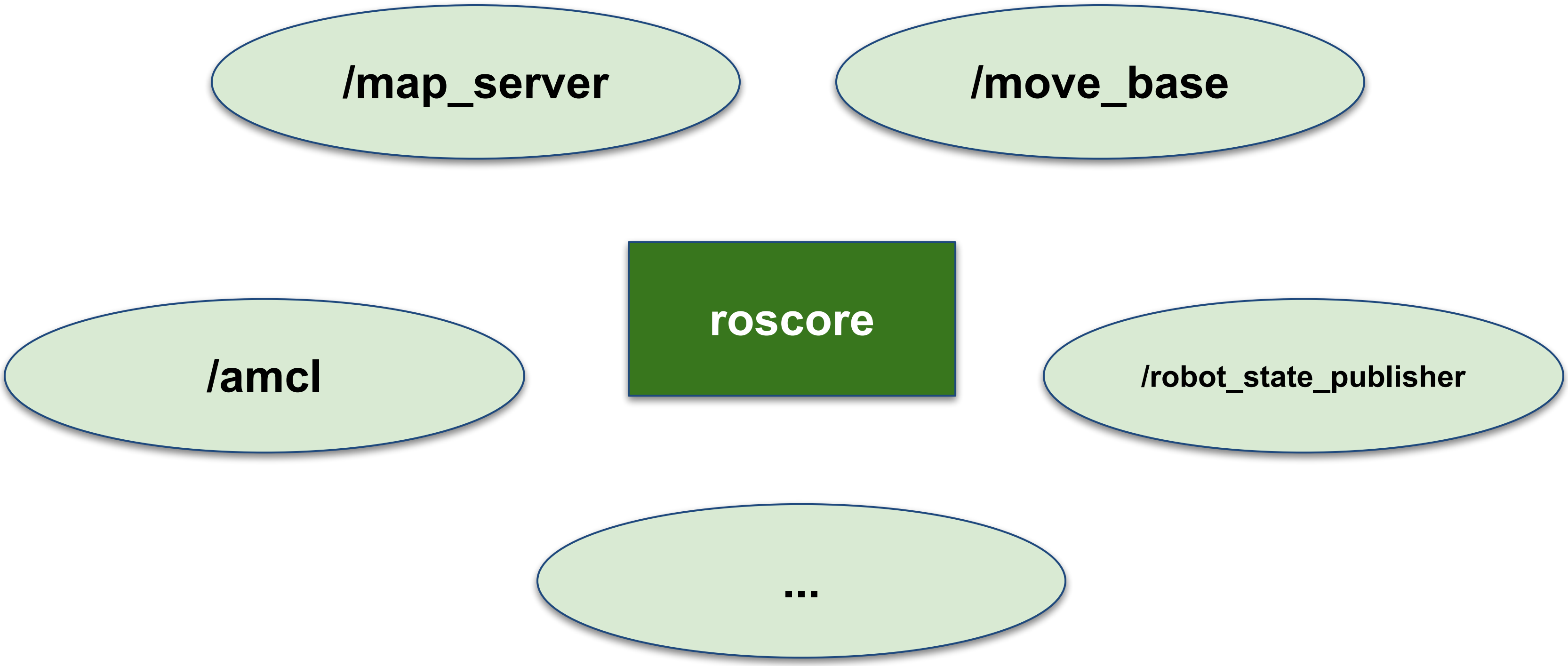

https://bit.ly/35fFJfM

Architectural Styles vs. Design Patterns



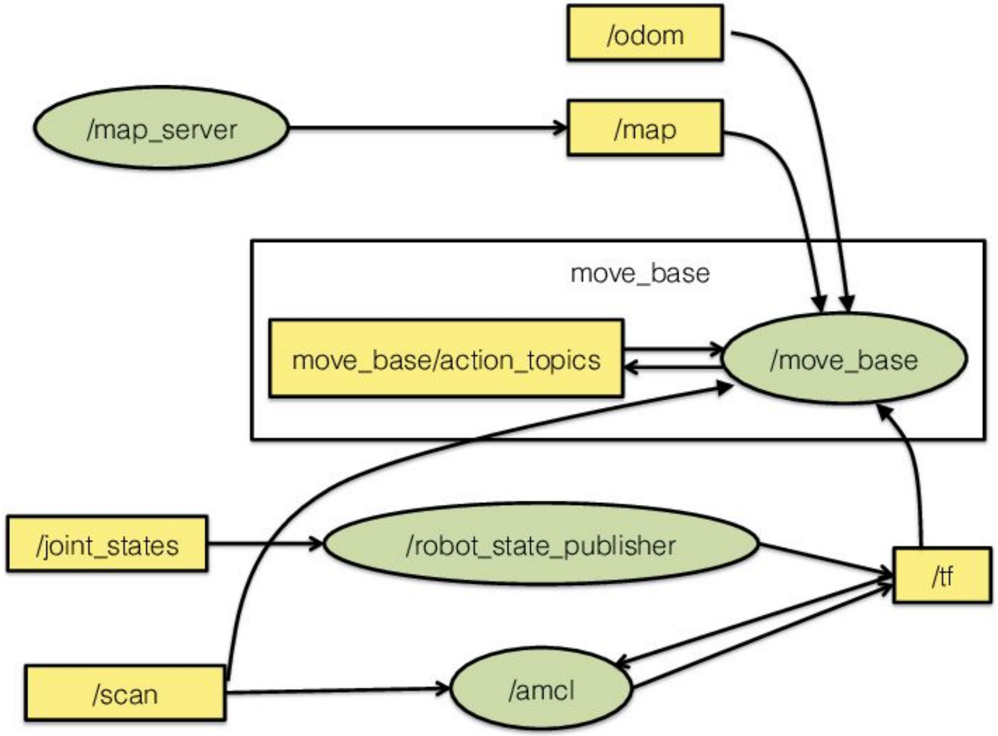


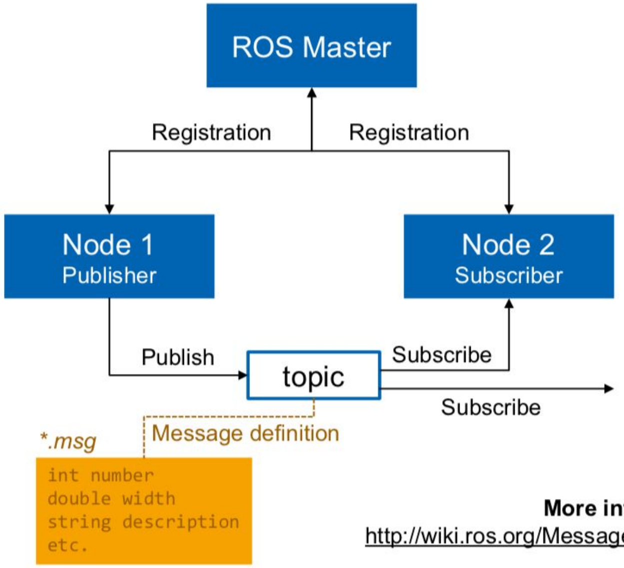
ROS Graph



institute for SOFTWARE RESEARCH

ROS: Publish-Subscribe Architecture

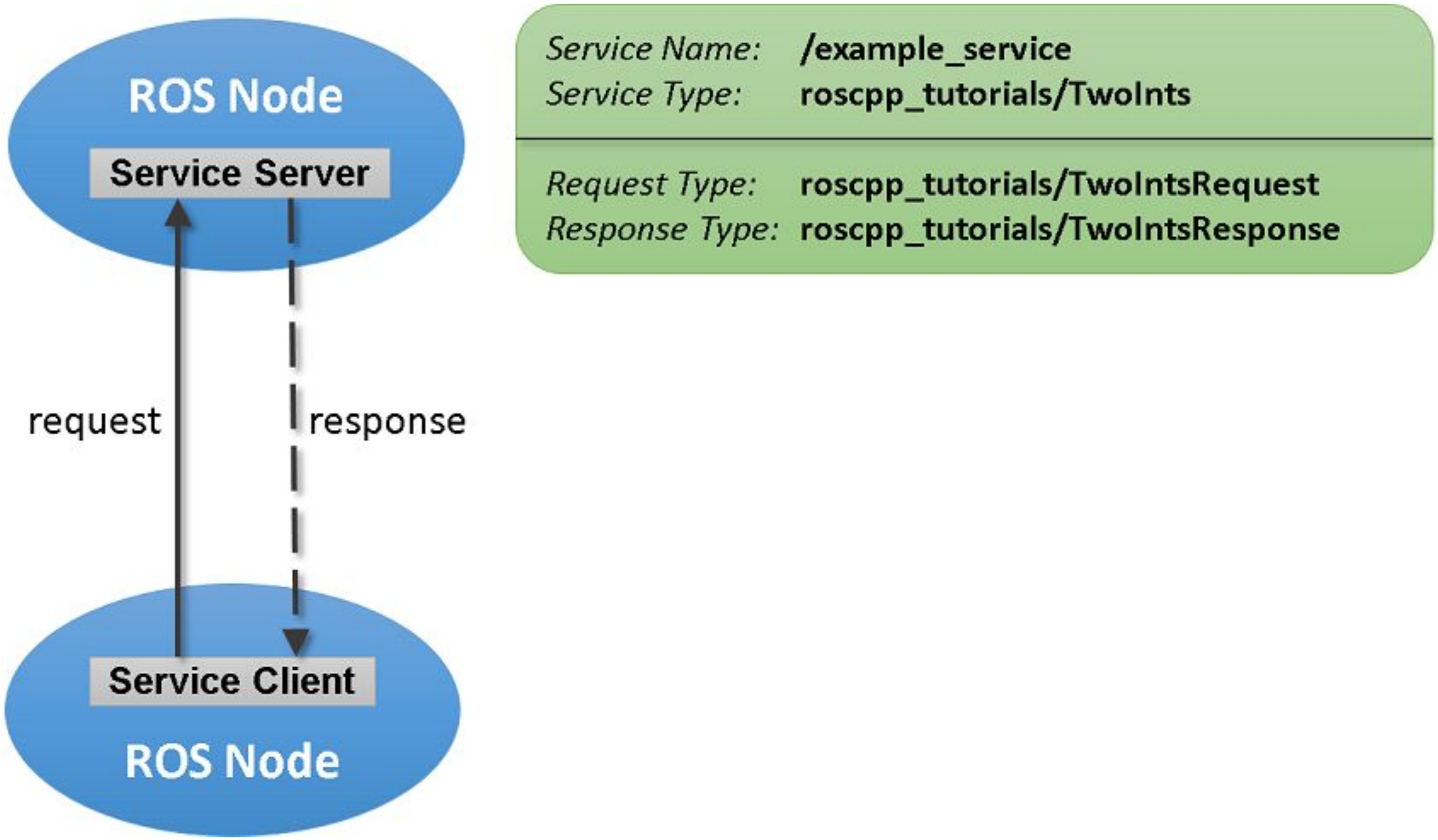




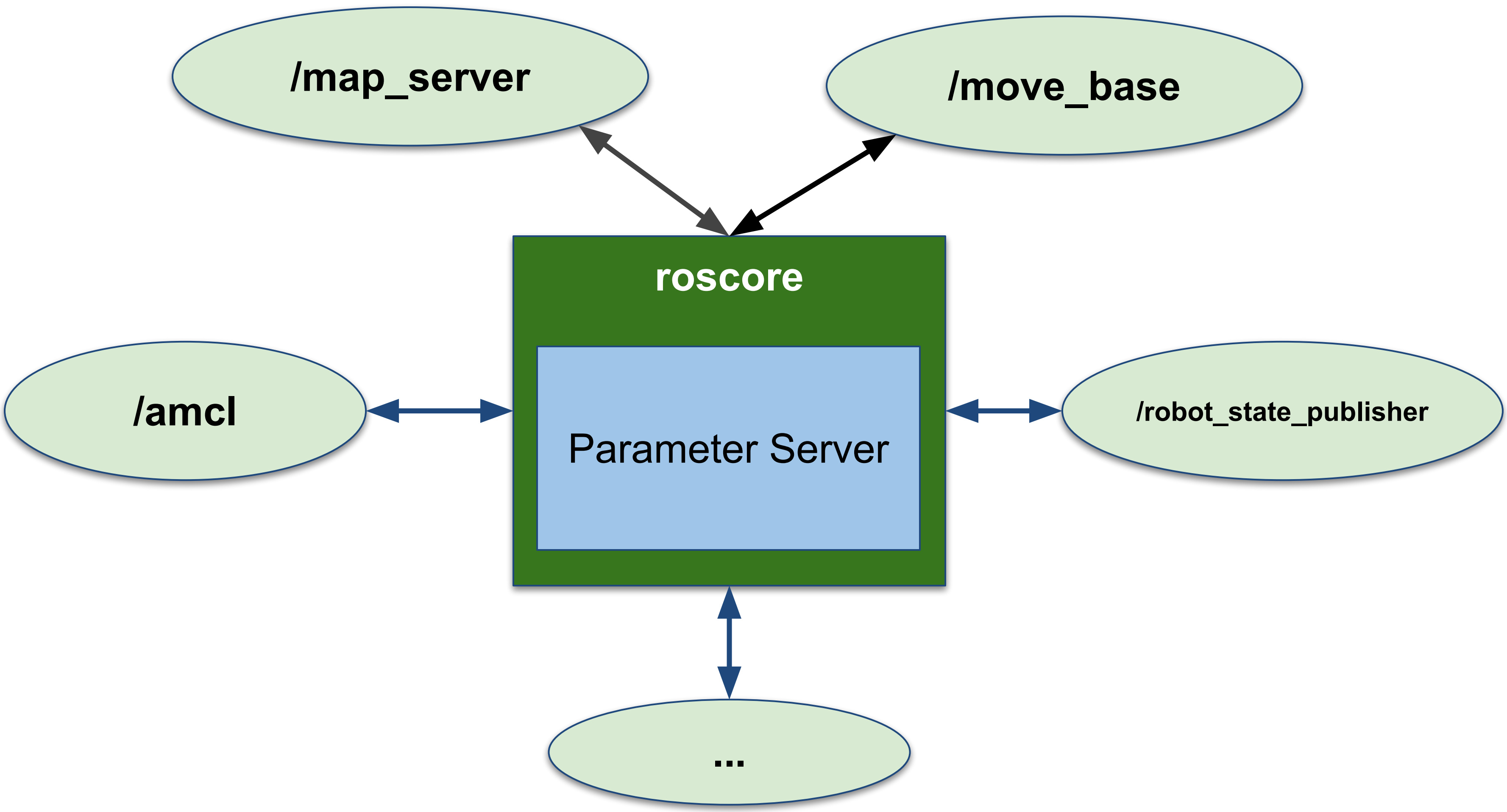
More info
http://wiki.ros.org/Messages

https://www.researchgate.net/profile/Anis_Koubaa/publication/309668701/figure/fig1/AS:424638028226561@1478253003400/Example-of-a-ROS-Computation-Graph-An-ellipse-represents-a-node-and-a-rectangle.png
https://miro.medium.com/max/814/1*tQs9gRoM5SePPvQProhlbg.png

ROS: Service Calls (Remote Procedure Calls)



ROS: Parameter Server



The evolution of ROS1 to ROS2

ROS

- **Single point of failure (roscore server)**
- Designed for researchers
- Assumes excellent network connectivity.
- Hard to build multi-robot systems
- Lack of security
- No built-in real-time control support
- ...

2

- **No need for roscore! Uses dynamic discovery.**
- Designed for production
- Operates with degraded network connectivity
- Built for multi-robot systems
- Secure communications over SSL
- Uses new technologies for a smaller implementation (e.g., DDS, Protocol Buffers).
- ...

Summary

- Robots are increasingly important to our everyday lives.
- Making sure that robotics software is well designed and tested is essential.
- ROS is an evolving software framework and ecosystem for robotics development.
  - Combination of architectural styles.
  - ROS1 was confined by assumptions, and so ROS2 was born.
- 17-313 jumps from small-to-medium-scale software design to large-scale software development in the wild.
  - Requirements, quality assurance, process, machine learning, large-scale software design, economics