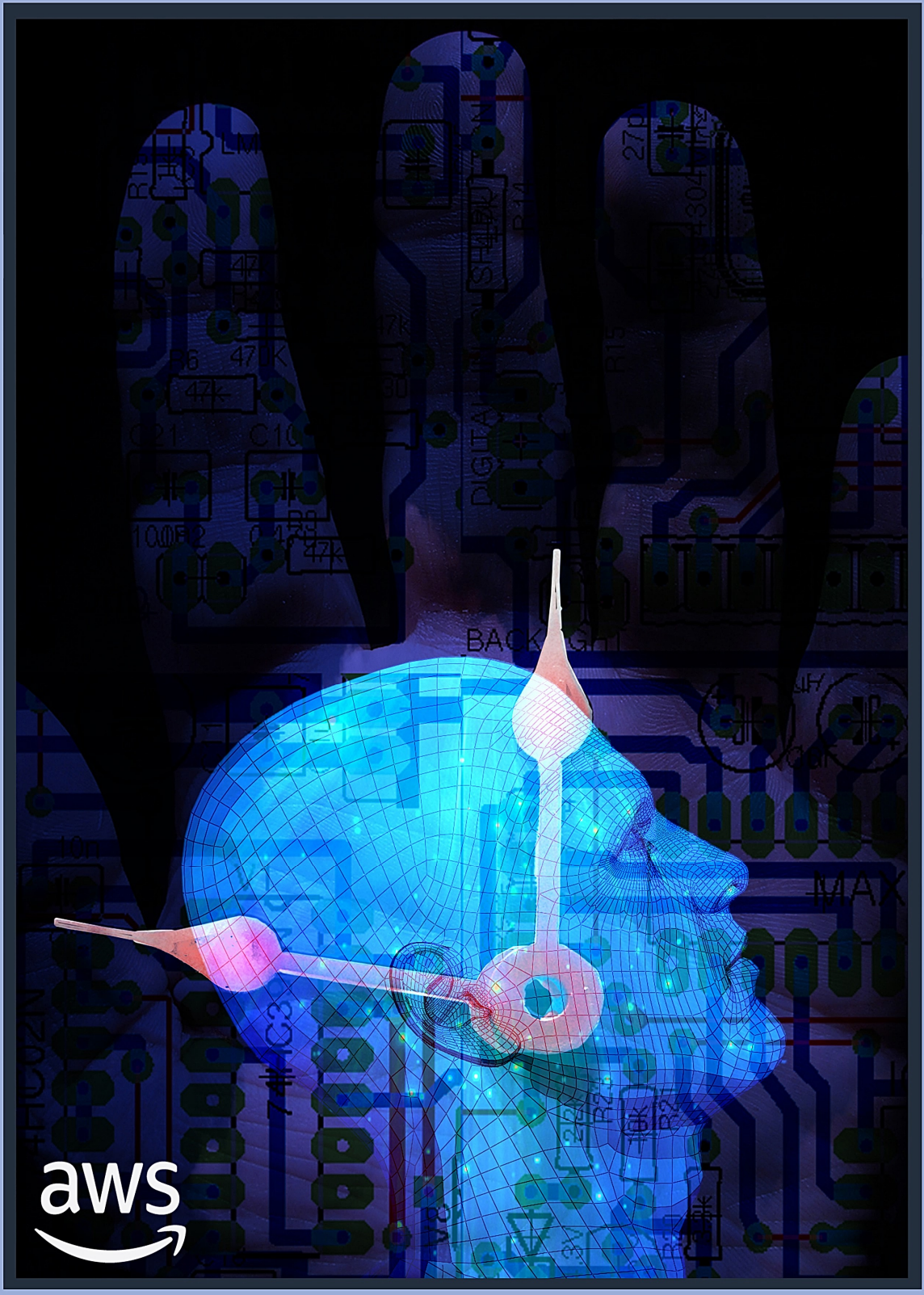
AWS Architecture Monthly



September 2020 **Robotics**

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers, or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Editor's note

This month's issue is all about robotics. Discover why iRobot, the maker of your favorite (though maybe not your pet's favorite) little robot vacuum, decided to move its mission-critical platform to the serverless architecture of AWS. Learn how and why you sometimes need to test in a virtual environment instead of a physical one. You'll also have the opportunity to hear from technical experts from across the robotics industry who came together for the AWS Cloud Robotics Summit in August.

Our expert this month, Matt Hansen (who has dreamed of building robots since he was a teen), gives us his outlook for the industry and explains why cloud will be an essential part of that.

We hope you'll find this edition of Architecture Monthly useful, and the team would like your feedback. Please give us a star rating and your comments on the Amazon Kindle page. You can view past issues and reach out to aws-architecture-monthly@amazon.com anytime with your questions and comments.

We're asking you to take a 10-question survey about your experiences with this magazine. The survey is hosted by an external company (Qualtrics), so the below survey button doesn't lead to our website. Please note that AWS will own the data gathered from this survey, and we will not share the results we collect with survey respondents. Your responses to this survey will be subject to Amazon's Privacy Notice. Please take a few moments to give us your opinions.

Take the Survey »

In September's Robotics issue:

- **Ask an Expert**: Matt Hansen, Principle Solutions Architect
- **Blog**: Testing a PR2 Robot in a Simulated Hospital
- **Case Study**: iRobot
- **Blog**: Introduction to Automatic Testing of Robotics Applications
- **Case Study**: Multiply Labs Uses AWS RoboMaker to Manufacture Individualized Medicines
- **Demos & Videos**: AWS Cloud Robotics Summit (August 18-19, 2020)
- **Related Videos**: iRobot and ZS Associates

Annik Stahl, Managing Editor

What are the general architecture pattern trends for the robotics industry?

Robots generally have a common top-level architecture. First, they gather data from sensors such as cameras, Light Detection and Ranging technology (LiDARs), wheel encoders, and accelerometers. After feeding this data into some high level processing/detection logic, some perception modeling occurs and the robot plans what actions to perform. Then, the robot sends commands to its actuators to move in some way. These actions are continuous and operate in parallel, as a closed loop feedback system.

Because of this need to have continuous parallel components, most modern robotics architectures are distributed "nodes" or processes running in parallel, communicating via a publish/subscribe design pattern. This has been a long running trend in robotics, particularly during the last 15-20 years, as the growth of the modern multi-core microprocessors and graphics processors has made the required performance achievable without special hardware.

Another technology that has accelerated this architectural pattern is Robot Operating System (ROS), an open source software built to make this type of architecture easy. It provides easy-to-use APIs for creating publishers, subscribers, services, and actions, common messages for interfacing between nodes, development tools, and even interfaces for simulating robots in 3D. ROS has been available for about 10 years, and it has become the standard software development kit for everyone from high school students to NASA.

When putting together an AWS architecture to solve business problems specifically for robotics, do you have to think at all differently?

Even though robots are very different from typical business applications, the business problems in robotics are similar to those in other areas. For example, software development for robotics requires a high quality, continuous integration (CI) system with quality test content to ensure the software is working. We can achieve this through a combination of unit and system tests integrated into a CI system such as AWS Code Pipeline, where the system tests are 3D robot simulations running in AWS RoboMaker. Another example is for code deployment to the robots in the field using over-the-air (OTA) deployment tools, such as AWS IOT Greengrass. In this way, developing software and deploying it to robots is similar to other IOT software workflows.

Do you see different trends in robotics in cloud versus on-premises?

Yes, the trend now is to connect robot fleets to the cloud for monitoring and logging telemetrics, remote troubleshooting, tele-operation, and adding features like voice and facial recognition. While some of these capabilities are possible on-premises, most are improved or made possible from remote locations by connecting the robot to the cloud. This trend is fairly new (in the last 2-4 years), but it is growing quickly and will likely become standard for robots in industry.

What's your outlook for the robotics industry and, what role will cloud play in future development efforts?

The robotics industry is still in its early stages of introduction and early growth, and analysts have forecasted accelerating growth through the rest of this century. In my opinion, I believe we're at the beginning of the exponential growth curve, similar to what the PC was in 1980 and smart phones were in the early 2000s. I believe that in 20 years, robots will be in homes, hospitals, restaurants, and everywhere people live, work, or play. Cloud will be an essential part of that, as all those robots will be connected to each other and use the power of the cloud to learn new skills, communicate status, and to receive their missions and goals.

Anything else you'd like to add?

It's a very exciting time to be in robotics. Developing robots for everyone has been a dream of mine since I was a teenager, so I feel like I'm living my dream now. The technologies have reached enough maturity that now the dream of having autonomous robots in our daily lives is fast approaching, and I wake every day excited to learn what's new. I really love my job as a Principal Solutions Architect for AWS Robotics and look forward to our future growth together.

About our expert

Matt Hansen is a Principal Solutions Architect specializing in Robotics at Amazon Web Services (AWS). Prior to joining AWS, Matt had 5 years working with ROS and ROS 2 in Intel's Open Source Robotics team, where he led the ROS 2 Navigation2 project and was an original member of the ROS 2 Technical Steering Committee. He is an Oregon native and has an MS in Electrical Engineering from Portland State University.

By Matt Hansen

Read online

Background

Nowadays, risk of spreading disease is a key concern in hospitals, where doctors, nurses and other caregivers are on the front lines helping patients. Hospitals have started using robots in daily operations such as contactless delivery and room disinfection to reduce risk of spreading disease. As the need for robots in healthcare grows, better tools are needed to build, test, and deploy robotics applications quickly and safely.

Introduction

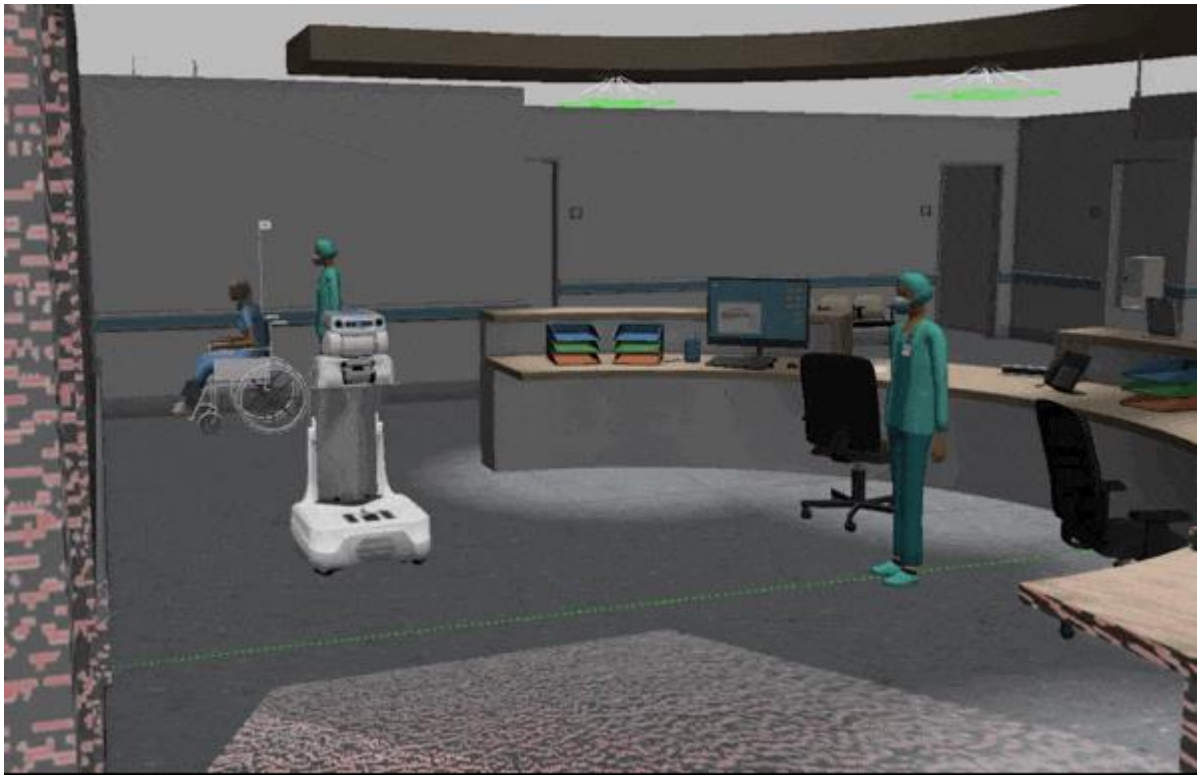
Testing robots in a physical environment is time consuming, and testing new code in a hospital environment can pose a safety risk due to the unpredictability of untested code. Testing in a virtual environment, or a simulated world, increases test coverage, reduces safety risk, and decreases development time. However, creating virtual worlds for simulation is costly, time consuming, and requires specialized skills in 3D modeling. For that reason, Amazon Web Services (AWS) has developed a Gazebo simulated Hospital World and published it as open source so that robotics companies within the Healthcare industry can more easily test their robots in a simulated hospital environment. In this blog, I will provide an overview of the hospital and share my experience using it to test a PR2 robot, including software failures that I encountered that you may find helpful.

In order to test and demonstrate the features of the Hospital World, I wanted to use a robot that moves and looks like I would expect a hospital based robot to move and look if I were to develop one myself. I researched available options from the open-source community, searching for robots that had existing simulation packages, and the one that achieved these needs was the PR2 robot. The PR2 was developed by Willow Garage in conjunction with ROS as a development and research platform. It's great for simulation, running navigation and SLAM, and it's fun to play with!

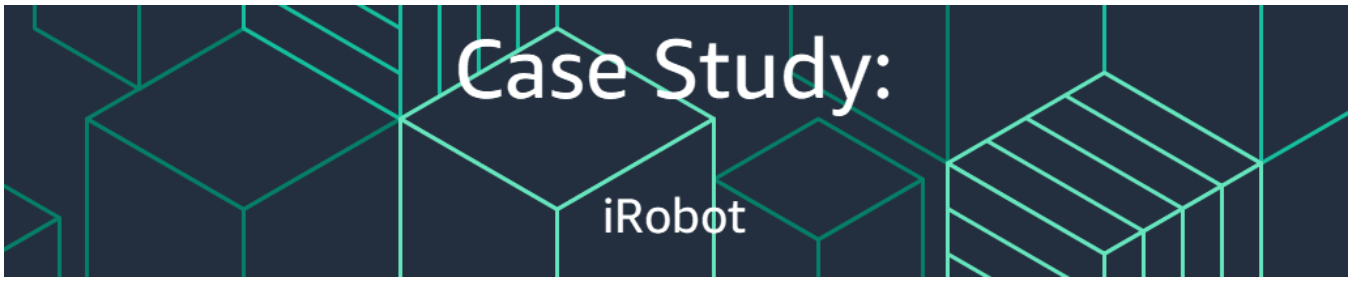
Overview

Now that we have background on why this matters and what robot was chosen for testing, we will dive into these three areas:

1. A ‘quick tour’ of the Hospital World
2. Testing the PR2 robot in the Hospital World
3. Getting started with the Hospital World



To see the all Hospital World graphics and simulations, read the full post online.



Read online

iRobot, a leading global consumer robot company, designs and builds robots that empower people to do more both inside and outside the home. iRobot created the home-cleaning robot category with the introduction of its Roomba Vacuuming Robot in 2002. Today, iRobot is a global enterprise that has sold more than 20 million robots worldwide. iRobot's product line, including the Roomba and the Braava family of mopping robots, features proprietary technologies and advanced concepts in cleaning, mapping, and navigation. iRobot engineers are building an ecosystem of robots and data to enable the smart home.

> *'The AWS serverless architecture and the ease of use of the AWS services inside it free up developer time to produce business value."*
>
> **Ben Kehoe**
> Cloud Robotics Research Scientist, iRobot

The challenge

The first Amazon Prime Day was a good one for iRobot. On that July day in 2015, the company sold 14,000 of its Roomba robotic vacuums. This one-day sales number underlined the reputation for innovation and value iRobot had built among its customers. It also underlined a challenge iRobot would face just a few months later, in September, when the company would release its first internet-connected Roomba vacuums.

"Roomba vacuums are popular Prime Day purchases and holiday gifts, and people want to try them out as soon as possible," says Ben Kehoe, a cloud robotics research scientist at iRobot. "So there are certain days, particularly after big sales events like Prime Day, when a large number of customers send their newly purchased Roomba vacuums on their first cleaning missions."

In July 2015, spikes in new Roomba use did not directly impact the company's day-to-day operations. But starting that September, large numbers of people trying out the new connected Roomba vacuums would result in large volumes of traffic through the iRobot HOME App, the mobile app customers would be using to set up and control their connected robots. In other words, iRobot, which had so far operated primarily as a hardware vendor, was about to bet its central line of business on its ability to run a high-availability, customer-facing cloud application and an Internet of Things (IoT) backend platform.

It was a gamble that paid off. Today, iRobot reports that connected Roomba vacuums operate in more than 60 countries, with total sales of connected robots projected to reach more than 2 million by the end of 2017.

iRobot isn't stopping there. Not only is the company sending its connected Roomba vacuums to save us from the drudgery of vacuuming; the company believes these robots can one day help solve a problem that has so far prevented the IoT industry from delivering on the true promise of smart-home technologies: a house that simply knows what to do and reacts accordingly.

First priority: customer focus

As always, however, iRobot remains focused on its central value proposition: leading-edge robots that relieve customers from menial chores and give them time for what's more important. That customer focus is why iRobot started offering connected Roomba vacuums. "Customers are demanding easier ways to interact with a growing number of products and technologies throughout the home," says Kehoe. "Cloud connectivity provides Roomba customers with even more convenience and control, so they can use their phones to manage their Roomba, wherever and whenever it's convenient."

In 2013, as it planned the platform that would be the foundation of this new offering, iRobot decided to build it in the public cloud. For the initial launch, iRobot worked with a vendor that provided a turnkey solution for launching the first cloud-connected Roomba in September 2015. However, as iRobot began to build out its family of connected Roomba vacuums—and as the sheer number of connected customers and services quickly multiplied—iRobot recognized it needed a solution that could scale more quickly and allowed for more direct control.

"Following the launch of our first connected product, it became increasingly clear that we couldn't achieve the scale and extensibility we needed with the turnkey solution we were using," says Kehoe.

Why Amazon Web Services

iRobot decided to move its mission-critical platform to the Amazon Web Services (AWS) Cloud. "The AWS Cloud offered an essential combination of scalability, global availability, and breadth of services," says Kehoe. "What really grabbed our attention is that AWS offers powerful tools and integration capabilities that enable us to use a serverless architecture that saves us the headaches of learning to scale."

To run the web applications that connect to the new Wi-Fi-connected Roomba vacuums, iRobot is using about 25 AWS services. At the core of the iRobot platform are AWS Lambda and the AWS IoT platform. AWS IoT, which can process trillions of messages between

billions of devices and AWS or other endpoints, provides the connectivity layer between the robots and the iRobot cloud platform. AWS Lambda runs code in response to events to provide function-based compute services for the serverless backend that powers the iRobot cloud application. The solution also uses Amazon Kinesis, which ingests real-time data streams from robots, and the Amazon API Gateway, which can publish APIs and secure them at any scale.

By using a serverless architecture based on AWS IoT and AWS Lambda, iRobot is able to keep the cost of the cloud platform low, avoid the need for subscription services, and manage the solution with fewer than 10 people. "Running on an AWS serverless architecture lets us focus on code and customers rather than operations," says Kehoe. "The AWS serverless architecture and the ease of use of the AWS services inside it free up developer time to produce business value."

The benefits

Gartner, Inc., forecasts that there will be 20.4 billion "connected things" in use worldwide by 2020. In other words, the number of homes containing connected refrigerators, lights, HVAC systems, and security alarms is only going to grow. But iRobot argues that the value of all this connectedness will remain out of reach without a crucial category of data: up-to-date mapping information about the houses where the devices reside.

"To achieve a seamless smart-home experience in which the home, and the smart devices within it, responds to our daily needs autonomously, two things must happen," says Kehoe. "First, the burden of programming devices must be removed from the consumer. And second, the home needs to understand itself: what the layout of the home is, the location and purposes of each room, and where the home's various smart devices are located."

According to Kehoe, connected robots like the Roomba hold the key. iRobot's Roomba 900 Series vacuums, for example, use visual simultaneous localization and mapping (vSLAM) technology to navigate an entire level of a home. As the Roomba vacuums, it builds a map of the home, and it is this map that could provide the foundational information needed to help a smart home understand itself and enable truly seamless, autonomous home automation. While today's Roomba is creating maps to clean more effectively, future generations of Roomba vacuums could combine these maps with additional sensors and supportive cloud services to manage the myriad smart devices within a home.

As straightforward as this vision is to state, it's quite challenging to achieve, but Kehoe says it's possible in the AWS Cloud. "The information pathways for something like this are incredibly complicated, security for customer information is crucial, and execution won't ultimately be possible without powerful artificial-intelligence and data-analysis tools. That's

why we're building all this in the AWS Cloud. Amazon Web Services is the key to making this vision a reality."

Read online

Blog:
Introduction to Automatic Testing of Robotics Applications

By Pulkit Kapur

Read online

Robotics applications such as warehouse logistics, automated driving, and factory automation require safe, error-free operation in dynamic real-world scenarios. Automated testing for these applications saves precious debugging time and helps improve the overall quality and safety of the software. This blog will explore best practices for testing and validation of robotics and autonomous systems running Robot Operating System (ROS) software. We will review how developers can increase their feature velocity and reduce errors by using simulation-based testing. We will focus on functional testing, which means testing whether the software provides expected results based on the design specification. Robotics software is multi-disciplinary with components for perception, planning, and controls. A small change in one part of the code base can adversely affect another component, causing a software bug. Software testing in robotics also requires testing for a wide variety of situations the robot encounters, including edge cases and unsafe conditions.

When testing robotics systems, a common approach is to use a V-model to verify that software meets the requirements. As shown in Figure 1, the left branch of V-model begins with system-level requirements (subsystems and components), which leads to high-level design and detailed design with simulation. Simulation is important to make sure that the design meets the requirements. The tip of the V-model is the coding or implementation phase to convert the design into code implementation. The right branch represents the testing of the components, starting with unit tests in code to integration tests of different subsystems. Verification tasks on the right branch are performed on the embedded software and target device.

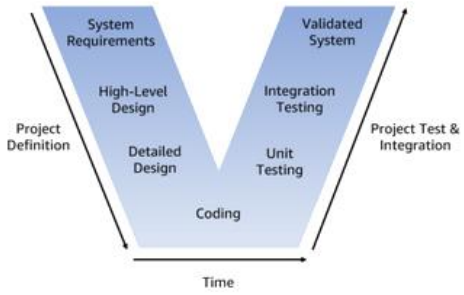


Figure 1: The V software testing framework

Three types of functional tests commonly used in robotics application development are: unit tests, integration tests, and regression tests. Together these tests can be used to verify and validate the robustness of the robotic software. Unit tests verify a specific component or module of your ROS application individually. Common tools for unit tests in ROS include

unit test, gtest, and pytests. Integration tests or system tests verify interactions between modules (such as multiple ROS nodes across libraries) making sure they work as designed and expected. Development teams use integration tests to find race conditions and deadlocks and to make sure that software components are fault tolerant and failure resistant. ROSTest is a popular tool to include functional tests into the native ROS build process. ROSTests have a .launch or .test extension. After testing your ROS function or system, an important step in software testing is running regression tests. This includes comparing the results of tests (unit tests or interaction tests) over time to harden software release and make sure that previous functionality remains intact.

Verifying a robot will behave as expected in a dynamic real-world environment is a complicated and time consuming testing task for robotics developers. In cases where hardware is not available, developers must wait until the end of the development cycle for physical testing. This delay results in expensive debugging and refactoring required to catch early design-stage bugs. Developers can increase their feature velocity and test their designs by using a physics-based 3D simulation that mimics their real-world application environment. **Simulation testing during early stages of robot development is especially valuable when developers have no access to hardware or physical testing space.**

Simulation-based behavior tests

Running tests in a simulated virtual environment enables robot software developers to confirm their applications result with the correct robot behavior. Simulation-based testing can involve: debugging an algorithm during iterative development, testing a subsystem such as localization or object detection, or testing the functionality of the complete system such as navigating from start to goal. A combination of different simulation fidelities is employed in testing, low fidelity 2D simulation for quick visualization and debugging and high-fidelity 3D simulation for system level behavior tests. Using a variety of tests (unit, integration, and frequent regression testing on a periodic basis) increases the overall test coverage of the code base and prevent expensive software defects.

Read the full post online

Read online

Personalization is the future of medicine. Currently, daily drug regimens can be challenging and complex, with patients taking multiple prescription drugs to treat different medical issues. Multiply Labs strives to solve this problem by manufacturing a single daily capsule containing a patient's entire prescription—with a dosage individualized to a patient's needs. The company supplies robotic manufacturing-as-a-service for this customized medicine production. "Medicine is becoming more personalized, and we help companies enable that by using robots to create pharmaceuticals in individually sized batches," says Zack Bright, chief technology officer for Multiply Labs.

Because its software must comply with strict Good Manufacturing Practice (GMP) regulations from the U.S. Food and Drug Administration (FDA), Multiply Labs needed to securely and automatically deploy software to the robots filling pills. "The software configurations for our robotic manufacturing cells must be monitored very closely so we can ensure traceability for compliance reporting," Bright says. "However, we relied on inefficient processes that included manually filling out stacks of paperwork for each batch. We needed a more automated process if we were going to grow our business."

Using AWS to automate robot software deployment

Multiply Labs decided to move its robotic software environment to Amazon Web Services (AWS) and take advantage of AWS RoboMaker to automate deployment of updated code to its robot application, based on Robot Operating System (version ROS 2). "AWS RoboMaker was ideal for us because now we can manage nine robot systems in parallel from our quality control software," says Bright. Previously, the organization relied on insecure SSH tunnels and scripts to pull and compile code programmatically—a complex process that, he says, "wasted engineering time and talent."

Using AWS RoboMaker, Multiply Labs deploys its robotic application multiple times a day to its robot fleet. "We hadn't done this type of testing before," Bright says. "We ran smoke tests and some regression tests, which took weeks." Multiply Labs also plans to incorporate simulation testing of its robots in the pipeline. "Without simulation testing, we would have to shut down the manufacturing line to test. Simulations mean we can maximize uptime as a robotics-as-a-service company."

In addition to implementing AWS RoboMaker with its continuous integration pipeline, Multiply Labs also uses Amazon CloudWatch to log all versions of each drug manufacturing batch.

Achieving continuous compliance for regulated workloads

The company uses AWS CodeDeploy and AWS CodePipeline to set up and deploy software builds, and it relies on Amazon DynamoDB to store data. Its robots interact with servers that send ROS 2 batch data in and out of an Amazon DynamoDB database. "By storing the batch records digitally on Amazon DynamoDB, we have a repeatable and traceable way to store data," Bright says. "We can clearly show that an exact amount of a patient's medication was filled by a certain machine in a certain order."

With a GMP integration of Amazon DynamoDB and ROS 2, plus AWS RoboMaker, Multiply Labs can enable digital batch records to be made by robots. "Before, our paper records were hundreds of pages long and took hours to fill out and review by hand," says Bright. "We were lucky to handle two batches at a time. Our new stack enables dozens or, as we scale, hundreds of batches to be manufactured at once. The flexible storage schema in Amazon DynamoDB, with a rapidly scalable capacity, means we can handle this for all our customers and store and manage thousands of batches at once at full scale."

Using AWS, Multiply Labs can enable consistent, controlled software deployments with a validated image every time. "We can ensure the software used during manufacturing was deployed exactly per our gold standard by using AWS RoboMaker and AWS CodePipeline," says Bright. "We automatically load the software onto our robots, multiple times a day, immediately before the robot runtime. There's no chance the wrong version of the software is running or that anything could be tampered with. Because of AWS RoboMaker, we can achieve continuous compliance for our FDA-regulated workloads and demonstrate traceability."

Enabling personalized medicine

Multiply Labs delivers personalized medicine more easily by relying on AWS. "The best way to do so is by using robots to make customized pharmaceuticals in individualized batches," says Bright. "That requires a fully automated and parallelized system where robots are in perfect sync. Using AWS RoboMaker, we can centrally orchestrate each manufacturing cluster of robots, with 9–15 different robots all connected to AWS RoboMaker. Each robot has different tasks such as transporting, weighing, or sorting capsules and then packing and sealing them. As a result, we can deliver customized therapies to patients in a single pill. We are also ready to scale this to all our customers quickly. Using AWS RoboMaker, it will be easy for us to go from our first nine robots to our next 100."

Saving a year of development time

Using AWS RoboMaker to automate deployment, Multiply Labs can track, trace, and inspect robot software to a level that would normally require time-consuming work. “Creating a solution like this on our own would have taken us a year of development work,” says Bright. “It would not have been possible to do what we wanted to do without a tool like AWS RoboMaker.” Deployment is automated and integrated into the company’s continuous integration and continuous delivery (CI/CD) pipeline, so engineers can focus on robot programming, not deploying code. “This easily saves an hour or two a day for our engineers. Over the next year, this would effectively be tens of thousands of dollars saved.”

Multiply Labs will also meet its goal of providing a manufacturing-as-a-service model to pharmaceutical companies. “Having an AWS-based system allows us to deploy both hardware and software in a scalable way, so we can sell our manufacturing capacity as a service,” Bright says. “And we can deploy repeatedly, using separate instances to assure our customers that their data is protected and not seen by others.”

With these capabilities, Multiply Labs can support customers’ clinical trials. “We can repeat instances of our software, put it into a shipping container, and send it directly to a company starting a clinical trial,” says Bright. “It effectively looks the same to our customers who deploy our systems abroad or have us manage the robots in our facility. We can manage the software from our San Francisco headquarters, no matter where the trial is across the globe. Meanwhile, our customers can see that the batches are being fulfilled in real time whether the robots are on-site or in Multiply Labs’ facilities. Using AWS definitely helps us meet our strategic goals.”

Read online

During August 18-19, 2020, technical experts from across the robotics industry came together for a complimentary educational event. The program was designed to help you learn best practices and the latest technology for robotics application development. Check out the many sessions that were hosted by AWS Robotics engineers and solutions architects with guest speakers from the Open Robotics, ROS-Industrial Consortium, iRobot, and Labrador Robotics. We've chosen three of the most popular sessions to share with you.

Keynote

Roger Barga, GM for AWS Robotics, and Brian Gerkey, co-founder of Open Robotics, will walk you through the latest in cloud robotics and open source development, including insight into the Robot Operating System (ROS)/Gazebo Ignition release roadmap. The speakers will discuss the value of using the cloud, open source frameworks, and simulation to help builders increase development velocity, test more accurately, and get robots to market faster.

Watch the Keynote

Simplifying simulation worlds creation for Gazebo

Simulation testing can help make robotics development faster, safer, and higher quality. Building virtual worlds is required to test robotics applications in simulation, and the process can require months of time and substantial expense for robotics companies. In this session, we will walk you through how to use AWS RoboMaker WorldForge to simplify the process of getting a Gazebo-based simulation world up and running. Attend this session to learn how you can accelerate world creation, so you can shift more testing to simulation environments to increase your testing velocity, and expand your test coverage to improve your robotics application quality.

Watch the session

Common simulation use cases and patterns with AWS RoboMaker

In this session, we will introduce how to use AWS RoboMaker for simulation tests, particularly for fleet simulations with multiple robots. We will deep dive into how RoboMaker simulations work and how to run complex, horizontally scalable simulations. With this approach, customers are able to parameterize and test ROS applications with complex robot interaction scenarios. We will also show how you can automate these robot interaction test cases, and run simulations within a continuous integration (CI) pipeline.

Watch the session

iRobot: Vacuuming Up Microservices on AWS

Ben Kehoe, Cloud Robotics Research Scientist at iRobot, explains how they built a serverless solution to power microservices that scale to control millions of robots. You'll learn how they automate and optimize AWS Lambda function and Amazon API Gateway deployments with AWS CloudFormation and Swagger, plus how they inject information during the deployment process to decouple architecture details from the code.

http://amzn.to/AWS-robotics-iRobot-microservices

ZS Associates: Intelligent Process Automation Using AWS AI Services

Automation has been a key priority for ZS Associates' customers and its managed services business. ZS was one of the early adopters of AWS for scaling Robotics Process Automation (RPA), ensuring the right level of security, compliance, and governance. ZS is now innovating using AWS AI services like Amazon Transcribe and Amazon Comprehend, combining them with RPA to drive new and more impactful use cases.

https://amzn.to/AWS-robotics-ZS-Video

iRobot: Serverless Data Cataloging for Data Scientists

iRobot's serverless architecture for their connected robot fleet has been well documented in AWS re:Invent sessions. Recently, the iRobot data science team went into production streaming real-time event data from millions of robots into S3 and DynamoDB into a serverless ETL pipeline (SQS, Lambda, Kinesis, and Kinesis Firehose) and landing in an S3 bucket, where Glue jobs perform additional transformation. The resulting data is analyzed using Athena and used to populate fleet reports.

https://amzn.to/AWS-robotics-iRobot-video