Quality Assurance Initiatives for ROS Tool Development (Part II)

Zhoulai Fu
Sep 29, 2018
IT University of Copenhagen
TOKYO | HONDA RECALLS ACCORD, INSIGHT VEHICLES FOR SOFTWARE PROBLEM

TOKYO — Honda is recalling about 232,000 Accord vehicles and 2019 Insight hybrid cars in the U.S. for malfunctioning software for the rear camera.
An Important Goal: Making ROS (more) Reliable
Since 2017, we have been working on how to test ROS automatically

This talk:

- Dealing with ROS-specific challenges
- Reusing existing tools
- Demo from our on-going work
Challenge 1 in testing ROS: Lack of specification

- Turtlesim package
- Expected: draw square
- Bug: Turtle spins after 3/4

ROS bugs often occur "silently"
Challenge 2 in testing ROS: Lack of “good” test drivers

- Test drivers launch ROS components
- Good test drivers fail ROS components
Our solution: If you don’t have it, try to get it

❖ Challenge 1: Lack of specification ==> Use sanitizer
❖ Challenge 2: Lack of good test drivers ==> Use fuzzing
What is sanitizer?

- A build-in compiler option in GCC and Clang
- Automatically inject assertions: division-by-zero, array index out of bound

Demo available offline
What is fuzzing?

- 1989 experiment from Univ. Wisconsin
  - Pure random testing crashed 1/4 of the tested Unix utilities
- Fuzzing = **Smart** random testing
- Numerous bugs detected by fuzzing
- Microsoft, Google and many companies use fuzzing **daily**
Step 1: Inject specification with sanitizer

Spec-carrying ROS

Step 2: Crash the package with fuzzing
Demo from our on-going work
zhfu@ubuntu:~/catkin_ws$ rosservice call /turtle1/teleport_absolute 0 0 0
Conclusion on Tool Development part

❖ **Build** **reliable** ROS components
❖ **We reuse** existing solutions: sanitizer + fuzzing

Thank you! Questions?

Meet us at ROS-Industrial booth #10