Formalization and Runtime Verification of Invariants for Robotic Systems*

Ricardo Cordeiro[1], Alcides Fonseca[1], and Christopher Timperley[2]

[1] Faculdade de Ciências da Universidade de Lisboa, Lisboa, Portugal
[2] Carnegie Mellon University, Pittsburgh, PA

Abstract. Robotics has a big influence in today's society, so much that a potential failure in a robot may have extraordinary costs, not only financial, but can also cost lives. Current practices in robot testing are vast and involve such methods as simulations, log checking, or field tests. The frequent common denominator between these practices is the need for human visualization to determine the correctness of a given behavior. Automating this analysis could not only relieve this burden from a high-skilled engineer, but also allow for massive parallel executions of tests, that could potentially detect behavioral faults in the robots that would otherwise not be found due to human error or lack of time. We have developed a domain-specific language to specify properties of robotic systems in ROS. Specifications written by developers in this language can be compiled to a monitor ROS module, that will detect violations of those properties. We have used this language to express the temporal and positional properties of robots, and we have automated the monitoring of some behavioral violations of robots in relation to their state or events during a simulation.

Keywords: Robotics · Domain-specific language · Runtime Monitoring · Error detection · Automation.

1 Introduction

Robotics already have a great impact on our current society. Due to their broad practicality, the quality of software used by robots should be of extreme importance to us.

The Cyber-Physical systems of robots are non-deterministic and unreliable, mainly because robots interact directly with the real world. A sensor can return imprecise values since the environment itself can be very hard to predict. As a result, verifying whether a task or movement is correct can be hard for a system to conceive.

Current practices in testing robot software involve, field testing, simulation testing, logs checking, among others. The common denominator among these is that they require a human to analyze the behavior of the robot to determine whether the behavior is correct.

* Supported by organization x.

In the case of simulators, we can use the real value of objects' attributes in a simulation to compare with what the robot system perceives, but even so problems like what components to monitor and how to express them arise. Having a domain-specific language to specify a robotic system's properties can be useful or a burden depending on its complexity and accessibility.

This work has the objective of showing how a domain-specific language can be used to specify temporal and positional properties of robotic systems and monitor the simulation components associated with these properties.

The language allows describing a robotic system's properties in a somewhat simple and intuitive way, while at the same time still being able to express relevant temporal and positional arguments between robots and objects in the simulation. The language is supported by a compiler. The compiler translates the language to a monitoring mechanism. In this way, if a robotic system doesn't follow the properties defined by the user writing in the language, during execution, the compiler detects an anomaly and makes the analysis that the behavior of the robot is not consistent.

2 Language

3 Monitoring

References