UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Formalization and Runtime Verification of Invariants for Robotic Systems

Ricardo Jorge Dias Cordeiro

Mestrado em Engenharia Informática
Especialização em Interação e Conhecimento

Versão Provisória

Dissertação orientada por:
Alcides Miguel Cachulo Aguiar Fonseca

2021

Agradecimentos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius. Claritas est etiam processus dynamicus, qui sequitur mutationem consuetudium lectorum. Mirum est notare quam littera gothica, quam nunc putamus parum claram, anteposuerit litterarum formas humanitatis per seacula quarta decima et quinta decima. Eodem modo typi, qui nunc nobis videntur parum clari, fiant sollemnes in futurum. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.

Dedicatória

Resumo

Os documentos escritos em português devem ter um resumo em português e um resumo noutra língua comunitária que contenham até 300 palavras cada. Num trabalho final escrito em língua estrangeira, este deve ser acompanhado de um resumo em português entre 1200 e 1500 palavras.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius. Claritas est etiam processus dynamicus, qui sequitur mutationem consuetudium lectorum. Mirum est notare quam littera gothica, quam nunc putamus parum claram, anteposuerit litterarum formas humanitatis per seacula quarta decima et quinta decima. Eodem modo typi, qui nunc nobis videntur parum clari, fiant sollemnes in futurum. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.

Palavras-chave: Máximo de 5 palavras-chave separadas por vírgulas.

Abstract

Abstract in English here (max 300 words)!

Keywords: Maximum of 5 keywords separated by commas.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

Be it a robot arm in a car assembly line, your vacuum cleaner, or a cat-like robot to carry your food in a restaurant, 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 for us. Robot software as well as the techniques used to test their quality are very field-specific and different from the techniques employed in traditional Software Engineering. Automatic tests are barely used in robotics due to multiple factors: cost, complexity, hardware integration, among others [1]. The goal of this thesis is to overcome the challenges of automated testing in robotics, by providing developers with a usable alternative that allows detecting bugs with less effort through simulation.

1.1 Motivation

Today, robots are vastly used industrially (medicine, agriculture, etc.) or leisurely (contests, personal use, etc.). The tendency is for robot usage to keep growing at a global level. Robot tasks tend to be repetitive and rather specific. Therefore, Robot software also tends to be quite different from conventional software. The Cyber-Physical systems of robots are non-deterministic and unreliable, mainly because robots interact directly with their environment. 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 is hard for a robot to conceive.

Current practices on testing robot software are common among developers, including 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. If there was a tool that could make this decision, automated tests could be used more widely in robot systems. However, that is not the case as automatic tests are hardly used. Opening this door would mean an improvement in the quality of current and future robot software.

1.2 Problem Statement

The multiple challenges in robot testing have an influence on planning how to test a robot because there are tradeoffs among choices. While simulation-based tests are a promising approach for automation there is still distrust in the precision and validity of the results. This means that, despite being dangerous and sometimes expensive, real-life robot testing is still the main choice. Both in real-world testing or simulations, human supervising will most likely still be necessary. This is because identifying if a robot fulfills an expected behavior is very hard for the robot itself. For this reason, automatic tests in the robotics field are hardly reliable and hard to implement. The resulting product is a lack of quality in the software across projects. In short, right now in the field is manually costly to identify test scenarios and identify if the robot does what we want.

1.3 Objectives

This work has the objective of showing the potential of automatic tests in robotics and of simplifying their execution. With this in mind, we propose a mechanism that monitors a subset of the components of the robot during or after tests execution. These components aren't arbitrary but are defined with the help of a descriptive high-level language. Not only the components but the test scenario should be described in this language. With the description of this language, one should be able to detect and orchestrate relevant robot components associated with the testing scenario. The language should allow describing a robot property in a simple and intuitive way. This language will need to be supported by a compiler. The compiler should translate the language to a monitoring mechanism. In this way, if a robot doesn't follow the properties defined by the language, either during execution or a log analysis, the compiler will detect an anomaly in the normal behavior of the robot.

1.4 Contributions

The expected contributions of this thesis are below enumerated.

1. Definition of a descriptive high-level language to specify robots properties.
2. Implementation of compiler for the language that can be used for monitoring.
3. Evaluation of the expressive capability of the solution in real-world examples.

1.5 Structure of the document

The document is organized as follows:

- Section 1...
- Section 2...
- Section 3...

Chapter 2

Background & Related Work

Brief paragraph introducing the chapter.

The Robot Operating System (ROS) is an open-source framework with a vast collection of libraries and tools that help build robot software. ROS runs on Linux Ubuntu and provides an abstraction between hardware and software. ROS was built with the purpose of cross-collaboration, there are packages for almost everything and no need to reinvent the wheel. ROS is the most widely used tool for writing robot software. Companies like Sony, LG, Rapyuta Robotics, etc., rely on ROS to deliver their products.

Robot simulation is an essential tool for testing robots behavior. Gazebo started with the idea of a high-fidelity simulator to simulate robots in outdoor environments under varied conditions. Today it offers the ability to simulate numerous robots in complex distinct environments. Gazebo is an open-source 3D simulator that supports sensors simulation and actuators control under different physics engines.

Chapter 3

Proposed Approach

This is an example of a citation [2].

Appendix A

Bibliography

[1] Afsoon Afzal. A study on challenges of testing robotic systems.

[2] Alan M. Turing. Computing machinery and intelligence. In Margaret A. Boden, editor, *The Philosophy of Artificial Intelligence*, Oxford readings in philosophy, pages 40–66. Oxford University Press, 1990.