

Test-inspired runtime verification

Using a unit test-like specification syntax for runtime verification

Adam Renberg

The School of Computer Science and Communication (CSC),
KTH

March 17, 2013

TODO

- ▶ Change style...

Introduction

This is an exjobb. I did it at Valtech. My supervisors have been Narges Khakpour (KTH) and Erland Ranvinge (Valtech). My examiner is Johan Håstad.

Outline

Preface

Introduction

General Background

Problem Statement

Background

Verification

Runtime Verification

Unit Testing

pythonrv

Syntax

Instrumentation

f-pythonrv

Evaluation

Conclusions

Problem Statement

How can runtime verification specifications be written in a manner that uses the syntax of the target program's programming language, and resembles the structure of unit tests? Can we still give a formal semantics to the specification language, or a part of it? How can we bridge the spectrum of different approaches for verification, creating something in between the formal and informal techniques?

Background to Verification

What is Runtime Verification?

What is Unit Testing?

A

Evaluation

Using *pythonrv* on a real application: An intranet at Valtech.

Conclusions

List the questions from the problem statement again, but now with answers.

Future Work

List the future work section of the report.

Final Thoughts

Program size and complexity is increasing. Verification, formal or informal, can help with software development. Runtime verification is nice. Make it simpler to use, and readily available, and people will use it.