# Testing Quick Reference Handbooks in Simulators

Anthony Berg

21st April 2024

# Preface

## Abstract

This is an abstract.

## Declaration

I declare that this dissertation represents my own work except where otherwise stated.

## Acknowledgements

This is the acknowledgements.

# Contents

# Chapter 1

# Introduction

**Context**

- Designing Emergency Checklists is difficult
- Procedures in checklists must be tested in simulators [1], which usually means trained pilots test it, as the tests need to work consistently [2] (making sure it's not lengthy, concicse and gets critical procedures)
- Checklists are usually carried out in high workload environments, especially emergency ones

**Problem**

- Testing procedures in checklists are often neglected [1]
- There are some checklists that may not be fit for certain scenarios - e.g. ditching (water landing) checklist for US Airways Flight 1549 assumed at least one engine was running [3], but in this scenario, there were none
- Some checklists may make pilots 'stuck' - not widely implemented, could be fixed with 'opt out' points. e.g. US Airways 1549, plane below 3000ft, could have skip to later in the checklist to something like turn on APU, otherwise plane will have limited control [3].
- Checklists may take too long to carry out - Swissair 111

**Rationale**

- Test checklists in a simulated environment to find flaws in checklist for things like
  - Can be done in an amount of time that will not endanger aircraft
  - Provides reproducible results
  - Procedures will not endanger aircraft or crew further (Crew refering to Checklist Manifesto with the cargo door blowout)
- Results in being able to see where to improve checklists

# Chapter 2

# Background

# Chapter 3

# Design/Implementation

## 3.1 Abstraction

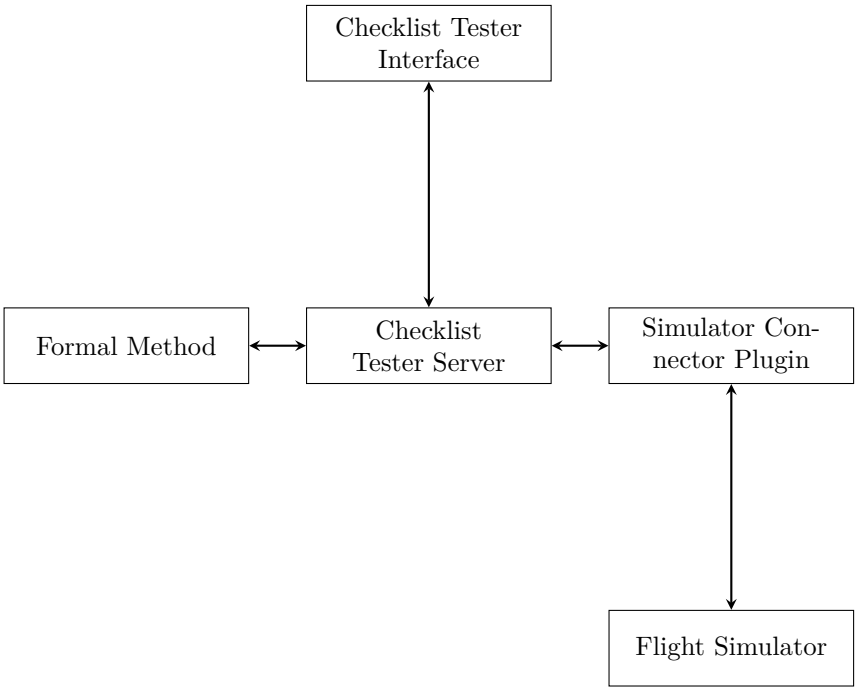


Figure 3.1: Abstract layout of components

## 3.2 Model

- Formal modelling is the heart of the logic for testing checklists
- Formal model created using VDM-SL
- It allows to test that the checklists have been completed in a proper manner - and that it is provable
- Model keeps track of
  - Aircraft state
  - Checklist state
- If an error were to occur in the model, this can be relayed that there was something wrong with running the test for the checklist, such as:

  - Procedure compromises integrity of aircraft
  - There is not enough time to complete the procedure
  - There is a contradiction with the steps of the checklist

## 3.3 Scenarios

- Use a Quick Reference Handbook (QRH) to find potential list of checklists to test
- Look at previous accident reports that had an incident related to checklists and test it with my tool to see if it will pick it up
- These previous accident reports can be good metrics to know what statistics to look out for

## 3.4 Decisions

- There would be around 3 main components to this tester
  - Formal Model
  - Flight Simulator plugin
  - Checklist Tester (to connect the formal model and flight simulator)
- As VDM-SL is being used, it uses VDMJ to parse the model [4]. This was a starting point for the tech stack, as VDMJ is also open source.
- VDMJ uses Java, therefore my language of choice was a language related to Java.

### 3.4.1 Formal Model

- There were a few ways of implementing the formal model into another application
- Some of these methods were provided by Overture [5]
  - RemoteControl interface
  - VDMTools API [6]
- However, both of these methods did not suit what was required as most of the documentation for RemoteControl was designed for the Overture Tool IDE. VDMTools may have handled the formal model differently
- The choice was to create a VDMJ wrapper, as the modules are available on Maven

# Chapter 4

# Results

## 4.1 Time Spent

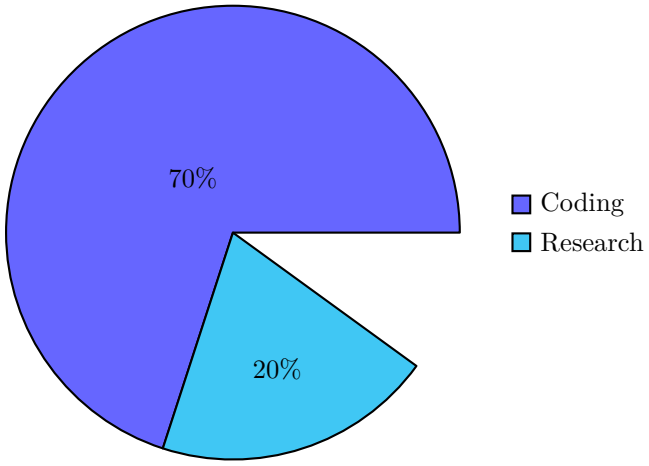


Figure 4.1: Time spent on ... **Improve wording**

# Chapter 5

# Conclusion

# References


[1] Immanuel Barshi, Robert Mauro, Asaf Degani et al. *Designing Flightdeck Procedures*. eng. Ames Research Center, Nov. 2016. URL: https://ntrs.nasa.gov/citations/20160013263.

[2] Atul Gawande. *The Checklist Manifesto: How To Get Things Right*. Main Edition. Profile Books, July 2010. ISBN: 9781846683145.

[3] National Tranportation Safety Board. *Loss of Thrust in Both Engines After Encountering a Flock of Birds and Subsequent Ditching on the Hudson River*. Technical Report PB2010-910403. May 2010. URL: https://www.ntsb.gov/investigations/Pages/DCA09MA026.aspx.

[4] Nick Battle. *VDMJ*. URL: https://github.com/nickbattle/vdmj (visited on 21/04/2024).

[5] Peter Gorm Larsen, Kenneth Lausdahl, Peter Jørgensen et al. *Overture VDM-10 Tool Support: User Guide*. TR-2010-02. Apr. 2013. Chap. 16, pp. 81–98. URL: https://raw.github.com/overturetool/documentation/editing/documentation/UserGuideOvertureIDE/OvertureIDEUserGuide.pdf.

[6] Kyushu University. *The VDM Toolbox API*. Version 1.0. 2016. URL: https://github.com/vdmtools/vdmtools/raw/stable/doc/api-man/ApiMan_a4E.pdf.

[7] Barbara Burian. 'Design Guidance for Emergency and Abnormal Checklists in Aviation'. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50 (Oct. 2006). DOI: 10.1177/154193120605000123.

[8] Quinn Kennedy, Joy Taylor, Daniel Heraldez et al. 'Intraindividual Variability in Basic Reaction Time Predicts Middle-Aged and Older Pilots' Flight Simulator Performance'. In: *The Journals of Gerontology: Series B* 68.4 (Oct. 2012), pp. 487–494. ISSN: 1079-5014. DOI: 10.1093/geronb/gbs090. eprint: https://academic.oup.com/psychsocgerontology/article-pdf/68/4/487/1520662/gbs090.pdf.

[9] Civil Aviation Authority. *Aircraft Emergencies: Considerations for air traffic controllers*. CAP745. Mar. 2005. URL: https://www.caa.co.uk/cap745.

[10] The Overture Project. *The Vienna Development Method*. URL: https://www.overturetool.org/method/ (visited on 23/02/2024).