A Theory of Refinement of Cyber-Physical-Systems into Implementations

Bachelor's Thesis of

Daniel H. Draper

at the Department of Informatics
Institute for Theorethical Informatics

Reviewer:	Prof. Dr. Bernhard Beckert
Second reviewer:	
Advisor:	Dr. rer. nat. Mattias Ulbrich

15. April 2015 – 15. August 2015

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

PLACE, DATE

Please replace with actual values

. . . . . .

(Daniel H. Draper)

Abstract

English abstract.

Zusammenfassung

Deutsche Zusammenfassung

Contents

List of Figures

List of Tables

1. Introduction

The following Bachelorthesis will try to formalize the following process: Replacing the abstract notion of the control program in a verified (by KeYmaera) Cyber-Physical-System (*CPS*) with an actual implementation through a form of Formal Refinement and being able to verify that the entire CPS still satisfys the required safety constraints, using both KeYmaera and KeY.

CPS are generally modelled as either Hybrid Automata or - Programs.(See ref. [**platzerb**]). Mostly, this means, that an abstract version of the discrete control program is modelled, often as a non-deterministic assignment of a control value (See app. A.1). To replace this non-deterministic assignment with an actual implementation a certain ”glue” or ”coupling” has to be found to translate discrete and real continous values into each other. To explain this process we will take a look at the following:

I: CPS Watertank (Example taken from KeYmaera) refined.

II: Introduction of Formalized used in both examples.

III: CPS Gear-Backlash (See ref. [**bla**]) refined.

2. Refinement of a concrete Example: CPS Controlled Watertank

2.0.1. Finding the concrete Control Value Assisgnment

2.0.2. Refining the original Hybrid Automata

2.0.3. Finding the correct Program Safety Condition

2.0.4. The (simple) Java Control Program

2.0.5. Finding the glue between Java and the Hybrid Model of the system

2.0.6. Verification based on KeYmaera

3. Evaluation

…

3.1. First Section

…

3.2. Second Section

…

3.3. Third Section

…

4. Conclusion

…

A. Appendix

A.1. Images

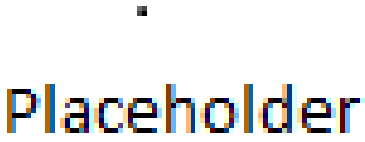

Figure A.1.: Watertank Hybrid Program and - Automata with Non-Deterministic Control Program Abstraction marked.