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

This 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 verified (by KeY) 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. [**platzer2010b**]). Mostly, this means, that an abstract version of the discrete control program is modelled, as a non-deterministic assignment of one or multiple 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. Glue in this case refers to a relation that encompasses both the discrete and real values. In certain cases (See chap. **??**), glue will be a concrete function, but no in general.

If we try to express the goal of this thesis in logic, we get equation 1.1. Here, ψ is the safety condition that acts both as a specification for our later implementation and the goal the program result is verified against. Glue is the aforementioned relation to be able to translate from the real into the discrete world and vice versa.

$$\begin{aligned} \textbf{If } (&\models [controlValue := *, ?\psi(controlValue) \dots]\alpha \text{ verified} \\ &\wedge \textit{"glue"(discreteVariables, continuousVariables)} \text{ verified} \\ \textbf{Then } &[controlValue := JavaProgram \dots]\alpha \text{ also verified} \end{aligned} \tag{1.1}$$

To explain this process we will take a look at the following:

I: Example-based Refinement on CPS Watertank (Example taken from KeYmaera).

II: Introduction of formalized process of using Refinement to gain a concrete implementation from a hybrid model.

III: Application of formalized process on example: CPS Gear-Backlash (See ref. [**bla**]).

1.1. Introducing Cyber-Physical-Systems

In this thesis we take a close look at **CPS**. These are systems in which a physical aspect or value is controlled by a computer (program). For example, an aircraft control system in which the computer exerts a form of speed control on the airplane would be a CPS.

In our case we take a closer look at the closely related notion of *Hybrid Systems*, in which discrete values (in the control program) and continuous values (in the physical

world) coexist. The difficulty in analyzing these kinds of systems stems from the "hybridness" of the systems: There is always some form of translation necessary to go from the program (discrete values) to the physical (continous reals) world.

two basic modelling approaches exist for hybrid systems: Hybrid Automata that are based on Non-Determnistic Finite Automata.

2. Example-based Refinement on CPS Watertank

To get a better understanding of the tasks involved in refining a hybrid model into a implementation with all necessary intermediate verification steps, we used one of the out-of-the-box examples provided in the KeYmaeratutorial [**keYmaera**].

2.1. Finding the concrete Control Value Assisgnment

In order to be able to apply Eq. 1.1 to this concrete example, the first challenge we faced was finding a spot in which a concrete control value is actually assigned. Taking a look at the Hybrid Automata describing the Watertank (See Fig. **??**),

2.2. Refining the original Hybrid Automata

2.3. Finding the correct Program Safety Condition

2.4. The (simple) Java Control Program

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

2.6. Verification based on KeYmaera

3. Introduction of formalized process of using Refinement to gain a concrete implementation from a hybrid model

What the Watertank example shows is the non-triviality of refining the hybrid model into an implementation and of the verification of all necessary parts. Overall it is obvious, that a formalized approach to the general problem presented in chapter 1 is necessary. In this chapter we present a possible formalized approach to the problem, that we deem feasible.

To aid readability we will now give an overview of the process without explanation, then detailing each step in the following sections.

1. Abstraction of the original hybrid model to better split actual control system "hook" and physical evolutions.

2. Finding the necessary safety condition of the control value for verification with KeYmaera.

3. Implementing control program according to safety condition as its specification and Verification by KeY.

4. Finding the correct "glue" between hybrid model and control program and its verification by KeYmaera.

5. Result evaluation: Was eq. **??** proven?

3.1. Abstracting original hybrid model to better split actual control system “hook” and physical evolutions

3.2. Finding the necessary safety condition of the control value for verification with KeYmaera

3.3. Implementing control program according to safety condition as its specification and Verification by KeY.

3.4. Finding the correct glue between hybrid model and control program and verifying it with KeYmaera

3.5. Evaluating results

4. Evaluation

…

4.1. First Section

…

4.2. Second Section

…

4.3. Third Section

…

5. Conclusion

…

A. Appendix

A.1. Images

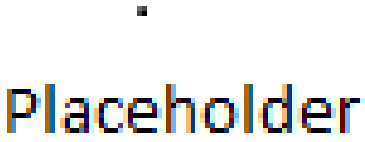

Figure A.1.: Watertank Hybrid Program and - Automata with Non-Deterministic Control Program Abstraction marked.