MODEL DRIVEN DEVELOPMENT FOR MEDICAL DEVICES IN AADL/BLESS AND SPARK/ADA: PCA PUMP PROTOTYPE

by

Jakub Jedryszek

B.S., Wroclaw University of Technology, Poland, 2012

B.A., Wroclaw University of Economics, Poland, 2012

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2014

Approved by:

Major Professor
John Hatcliff

Copyright

Abstract

Ada is a language is targeted at embedded and real-time systems.

SPARK is subset/superset (remove some funcionalities from Ada, but add contracts - in 2005, in 2014: only remove some stuff from Ada like no exceptions, no goto) of Ada with Code Contracts. It is designed for High-Assurance systems. It allows to prove corectness of program and its entities.

AADL (Architecture Analysis & Design Language) is modeling language for representing hardware and software. It is used for real-time, safety critical and embedded systems.

BLESS (Behavior Language for Embedded Systems with Software) is AADL annex sub-language defining behavior of components. The goal of BLESS is automatically-checked correctness proofs of AADL models of embedded electronic systems with software.

Nowadays, we have trend to generate code from models. The ultimate goal of research, which this thesis if part of, is to create AADL/BLESS to SPARK/Ada traslator. Ultimatelly there will be standarized AADL/BLESS models, which will be generating code base for developers extensions (like sceleton code for some Web Framework).

Medical Devices are very important part of High-Assurance systems.

This thesis propose mapping from AADL/BLESS to SPARK/Ada. As an example of Medical Device, PCA Pump (Patient Controlled Analgesia) is used. The foundation for this work is System Requirements for Integrated Clinical Environment Patient-Controlled Analgesia Infusion Pump (DRAFT 0.10.1) and AADL Models with BLESS annexes created by Brian Larson. Additionally, there was a contribution made in clarifying the requirements document and extending AADL models.

Table of Contents

List of Figures

List of Tables

Acknowledgments

//Stolen from Venkatesh

To be involved in a research project is a great experience.

I am thankful to Dr.Matthew Dwyer for offering me an opportunity (which I did grab!) to work in BANDERA group. I am thankful to Dr.John Hatcliff, my major professor, for being patient and persistent while guiding and mentoring me while working on this thesis. I am thankful for the numerous opportunities they both gave me to be involved in activities other than my thesis, and most of all for being enthusiastic and supportive. I am also thankful to Dr.Anindya Banerjee for providing useful inputs during the preparation of the thesis.

I am thankful to the entire BANDERA group for providing a great working envi- ronment. In particular, I am thankful to Hongjun Zheng for being a rigorous tester of my implementation. I am thankful to my friends for bearing with my antics, and making the long hot summers and bitter cold winters bearable.

I am thankful to my wife, Prathima, for being patient and supportive during my indulgence with my workstation for long hours and for bearing with my rather erratic feeding habit.

They say, save the best for last. Hence, lastly I would like to thank my mother, my father, and my elder sister for advising me in the ways of life and helping hold the helm during storms. Home is the first school is an adage in Kannada. I am indebted to my family for providing a great first school.

Dedication

For my family, mentors and all people who inspired me directly or indirectly in things I am doing.

I also dedicate this thesis to everyone who have supported me throughout the process.

Chapter 1

Introduction

The tale about software safety: why important, software everywhere, human life, etc. (info from 890).

1.1 Motivation

Why we need this model driven development. Why AADL? Why SPARK? Because is subset, which is easy to deal with it. In the future, when everything will be done (in case of proving perspective) in SPARK, it will (probably) be extended. Maybe finally, there will be no SPARK, but only Ada. Thus for now, SPARK is temporary subset of Ada for reasoning and corectness proving.

1.2 Contribution

Put all piecies together (SPARK, AADL, BLESS?, ICE, PCA Pump)

1.3 Organization

How many chapters etc.

Chapter 2

Background

General overview of all below things?

2.1 High-assurance systems

What software properties and requirements exists to ensure safety. Medical Devices as type of High-assurance systems.

2.2 SPARK/Ada

History of Ada: http://www.adahome.com/History/Steelman/intro.htm Ada fulfill US DOD requirements. Rationale of this language. Good for Software Verification etc. http://www.slideshare.net/A 2012 (slide 11: dev responsibility) GNAT Programming Studio. Tools for corectness proving.

2.2.1 Sireum Kiasan

Overview: symbolic execution, Pilar, Alir? Sireum Kiasan[**???**] is a tool, which use symbolic execution for finding possible paths in program. Plugin for GNAT Programming Studio. Plugin for Eclipse (but only SPARK 2005).

2.2.2 GNAT Prove

Overview

2.2.3 AUnit

Overview

2.3 AADL

Rationale of this language.

2.4 BLESS

How it fits into the picture. Why it was developed. Corectness prove in AADL + bahavior, from which we can generate SPARK/Ada code.

2.5 Integrated Clinical Environment

http://santos.cis.ksu.edu/MDCF/doc/ICE-Motivation.pdf http://santos.cis.ksu.edu/MDCF/doc/MDCF-Tutorial-Overview.pdf

2.6 PCA Pump

http://www.santoslab.org/pub/paper/LarsonEtAl13-PCA-Requirements-SEHC-preprint.pdf

2.7 AADL/BLESS to SPARK/Ada code generation

The ultimate goal of long term research, this thesis is part of. AADL-¿Ada BLESS-¿SPARK contracts + (eventually) behavior

2.7.1 Ocarina

Overview: AADL-¿Ada on PolyOrb, support for other languages?

2.7.2 Ramses

Overview. Not sure if it is useful at any point here?

Chapter 3

PCA Pump Prototype

Overview of PCA Pump and issues, which MDCF/ICE will solve.

3.1 PCA Pump Requirements Document

Selected use cases for implementation?

3.2 PCA Pump AADL/BLESS Models

Selected modules for implementation. Pictures etc.

3.3 BeagleBoard-XM

First step was create PCA Pump prototype on BeagleBoard-XM. No gnat compiler at the beginning. Then after cooperation with AdaCore, managed to run Ada program on BB (include source as appendix?). Only Ada (not SPARK), but working! Gnat compiler only for Linux Platform (for now).

3.4 PCA Pump Prototype Implementation

Issues: Ravenscar Profile, how to deal with different boluses (look at UMinn requirements and annotations for our doc). Look at annotated PCA Pump Req document.

Chapter 4

AADL/BLESS to SPARK/Ada translation

First step was to create mock (based on doc, aadl models and implemented PCA Pump).

4.1 Extention of exisitn PCA Pump AADL models

I added Subprograms etc. How I did it. Code examples.

4.2 AADL/BLESS to SPARK/Ada mapping

Table, how specific constructs (subset) in AADL/BLESS are translated to SPARK/Ada.

4.3 "DeusEx" translator

AADL/BLESS to SPARK/Ada translator in Scala. Main idea. Maybe at least create base: AADL-¿AST covertion?

Chapter 5

Summary

What I have done.

Chapter 6

Future work

What has to be done now.

Bibliography

Appendix A

Title for This Appendix

```
dm = 4
nm = 4
tm = 2
dimMAX = dm + 1;
part /: part[0] = {{}};
Do[part /: part[n] =
   Union[Flatten[
     Table[Sort[Join[{i}, #]] & /@ part[n - i], {i, 1, n}], 1]
   1, dimMAX}];
L = (Times @@ # &) /@ Map[c, part[dm], {2}];
```

Appendix B

Title for This Appendix