

Media Engineering and Technology Faculty
German University in Cairo



Behavioral Model Back-end Translator

2nd Interim Report

Author: Mohamed Hisham Soliman
Supervisor: Assoc. Prof. Georg Jung
Submission Date: 14 May, 2013

Media Engineering and Technology Faculty
German University in Cairo



Behavioral Model Back-end Translator

2nd Interim Report

Author: Mohamed Hisham Soliman
Supervisor: Assoc. Prof. Georg Jung
Submission Date: 14 May, 2013

This is to certify that:

- (i) the thesis comprises only my original work toward the Bachelor Degree
- (ii) due acknowledgement has been made in the text to all other material used

Mohamed Hisham Soliman
14 May, 2013

Acknowledgments

I would like to deeply thank my supervisor Prof. Georg Jung for his patience, support and guidance, because without his help I never would have understood many topics and technologies.

Big thanks to my class mates Omar Alfay and Ahmed Hodeeb for helping me through the whole process, especially with the eclipse plug-in development.

At last I must thank my friend Abdelrahman Mursi for helping me understand the Linear Temporal logic formulas, my friend Hadeer Younis for easing the process of writing the thesis for me.

Abstract

In this project, we build a translator for a given graphical behavioral description into the input syntax of a well known model-checking tool.

Contents

Acknowledgments	V
Abstract	VII
1 Introduction	1
2 Background	3
2.1 Technical	3
2.1.1 Eclipse Plug-in Development Environment (PDE)	3
2.1.2 Git repository	3
2.2 Theoretical	3
2.2.1 Spin	3
2.2.2 PROMELA	4
2.3 Related Work	4
2.3.1 Epispin eclipse plug-in	4
3 Tool Description	5
3.1 More detailed explanation for the tool	5
4 Tool Usage	7
4.1 How To Use The Plug-in	7
5 Conclusion	9
6 Future Work	11
Appendix	12
A Lists	13
List of Abbreviations	13
List of Figures	14
References	15

Chapter 1

Introduction

Graphical Behavioral models are diagrams that describe the behavior of a running process. A process is consisted of some states, each state is connected with a transition to the next state in the process. It's easier to describe a process using a drag and drop of some graphical figures than to describe it using lines of code. So our work in this thesis is to build a translator to automatically translate these models into the input syntax of a well-known model checker to be able to verify their correctness according to some user specifications.

Chapter 2

Background

In this chapter we present all the background information used while working on this thesis.

2.1 Technical

2.1.1 Eclipse Plug-in Development Environment (PDE)

The Plug-in Development Environment (PDE) provides tools to create, develop, test, debug, build and deploy Eclipse plug-ins, fragments, features, update sites and RCP products.[2]

2.1.2 Git repository

The project is hosted on the git repository: https://github.com/GeorgJung/Cadena_e.git for future work after submitting this thesis.

2.2 Theoretical

2.2.1 Spin

Spin is a generic verification system that supports the design and verification of asynchronous process systems.[4]

2.2.2 PROMELA

PROMELA(short for: PROcess MEta LAnguage) is the input language of Spin. It is used to describe the behavioral models for Spin to verify them.

PROMELA contains many features that are not found in mainstream programming languages. These features are intended to facilitate the construction of high-level models of distributed systems, The language supports, for instance, the specification non-deterministic control structures; it includes primitives for process creation, and a fairly rich set of primitives for inter-process communication.[3]

PROMELA derives many of its notational conventions from the C programming language. This includes, for instance, the syntax for boolean and arithmetic operators, for assignment (a single equals) and equality (a double equals), for variable and parameter declarations, variable initialization and comments, and the use of curly braces to indicate the beginning and end of program blocks. But there are also important differences, prompted by the focus in PROMELA on the construction of highlevel models of the interactions in distributed systems.[3]

A PROMELA model is constructed from three basic types of objects:

- Processes
- Data objects
- Message channels [3]

2.3 Related Work

2.3.1 Epispin eclipse plug-in

EpiSpin is an Eclipse plug-in for editing and verifying Promela models using the Spin model checker, developed using the Spoofax language workbench. It allows you to edit Promela models in Eclipse, using the well known Eclipse features.[1]

This plug-in is an easy tool to work with SPIN through it. It compiles the PROMELA code written on eclipse and generates the C code then compiles it for results. It also can generate the corresponding graph for the PROMELA code by converting it into something called Dot code then to the graph (state diagram) which will help in the debugging.

Chapter 3

Tool Description

This tool is a plug-in for eclipse. Therefore it's a java program. Simply a parser that parses an XML file describing the process features (i.e. nodes and transitions between nodes) then generates a PROMELA code accordingly for further verification with Spin.

3.1 More detailed explanation for the tool

some screenshots for the running tool and code snippets to explain the implementation process.

Chapter 4

Tool Usage

4.1 How To Use The Plug-in

A user manual with screenshots of how to use the plug-in when it's complete.

Chapter 5

Conclusion

Conclusion

Chapter 6

Future Work

As this tool is an eclipse plug-in, therefore it can be easily extended for further enhancements. A good extension for it would be to translate the PROMELA files back to the original model.

Another extension could be to allow editing the models form within the plug-in incase of errors in the model being translated.

Appendix

Appendix A

Lists

List of Figures

Bibliography

- [1] <http://epispin.ewi.tudelft.nl/>.
- [2] The Eclipse Foundation 2013. <http://www.eclipse.org/pde/>.
- [3] Gerard J. Holzmann. *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley Professional, 1st edition, 2003.
- [4] G.J. Holzmann. *Design and Validation of Computer Protocols*. Englewood Cliffs, N.J.: Prentice Hall, 1991.