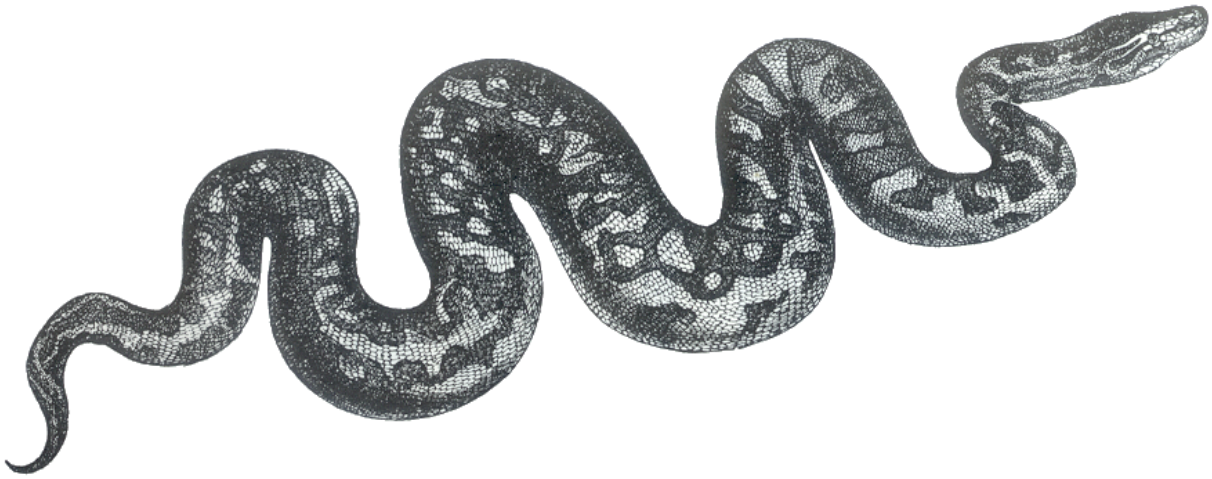
Department of Electrical & Electronic Engineering
The University of Auckland
New Zealand



Designing an application–specific programming language for mobile robots

Geoffrey Biggs

May 2007

Supervisor: Dr Bruce MacDonald

A thesis submitted in partial fulfillment of the requirements of Doctor of Philosophy in Engineering

Abstract

The process of programming mobile robots is improved by this work. The tools used for programming robot systems have not advanced significantly, while robots themselves are rapidly becoming more capable because of advances in computing power and sensor technology. Industrial robotics relies on simple programming tools usable by non-expert programmers, while robotics researchers tend to use general purpose languages designed for programming in other domains.

The task of developing a robot cannot be assumed to be identical to developing other software-based systems. The nature of robot programming is that there are different and additional challenges when programming a robot than when programming in other domains. A robot has many complex interfaces, must deal with regular and irregular events, real-time issues, large quantities of data, and the dangers of unknown conditions. Mobile robots move around and are capable of affecting everything in the environment. They are found in cluttered environments, rather than the carefully-controlled work spaces of industrial robots, increasing the risk to life and property and the complexity of the software.

An analysis of the process of developing robots provides insight into how robot programming environments can be improved to make the task of robot development easier. Three analyses have been performed: a task analysis, to determine the important components of the robot development process, a use case analysis, to determine what robot developers must do, and a requirements analysis, to determine the requirements of a robot programming environment. From these analyses, the important features of a robot development environment were found. They include features such as data types for data commonly found in robotics, semantics for managing reactivity, and debugging facilities such as simulators.

The analyses also found that the language is an important component of the programming environment. An application-specific language designed for robot programming is proposed as a solution for providing this component. Application-specific languages are designed for a particular domain of programming, allowing them to overcome the difficulties in that domain without concern for their usefulness in other domains. To test

the hypothesis that such a language would improve robot development a set of language extensions has been created. These extensions, named RADAR, provide explicit support for robotics. The prototype implementation uses the Python programming language as the base language.

RADAR provides support for two of the necessary features found in the analyses. The first is support for dimensioned data via a new primitive data type, ensuring all dimensioned data is consistent throughout a program. Dimensional analysis support is provided, allowing the safe mixing of data with compatible units, the creation of more complex units from simple single-dimension units, and built-in checking for errors in dimensioned data such as performing operations involving incompatible dimensioned data. For example, the data type will prevent the addition of distance and speed values. Several dimensional analysis systems have been developed for general purpose languages in the past. However, this is the first application of the concept specifically to robotics.

The second feature is semantics for managing reactivity. In RADAR, the principle of ease-of-use through simplicity is followed. Special objects represent events and responses, and a special syntax is used for both specifying these objects and managing the connections between them in response to the changing state of the program. This reduces programming complexity and time. There are many other languages for managing reactivity, both the more general languages, such as Esterel, and languages for robotics, such as TDL and Colbert. RADAR is simpler than the general languages, as it is aimed solely at the needs of robot developers. However, it takes a different approach to its design than other languages for reactivity in robotics. These are designed to provide support for a specific architecture or architecture style; RADAR is designed based on the needs of robot developers and so is architecture-independent.

RADAR's design philosophy is to provide robot-specific features with simple semantics. RADAR is designed to support what robot developers need to do with the language, rather than providing a special syntax for supporting a particular robot, architecture or other system. RADAR has been shown to provide an improvement in dimensioned data management and reactivity management for mobile robot programming. It increases the readability, writability and reliability of robot software, and can reduce programming and maintenance costs. RADAR shows that an application-specific approach to developing a robot programming language can improve the process of robot development.

Acknowledgements

When I first started my Phd, the three years I had ahead of me until my minimum submission date seemed like an eternity of work. Now that I have finished, the three and a half years it has taken seems like a roller coaster of activity squashed into a few memories. Needless to say, I would not have survived without the support of numerous people.

First on the list is my supervisor, Dr Bruce MacDonald, who has provided guidance and support since before I knew I wanted to do a PhD. Due to his support applying for a summer research scholarship at the end of my third undergraduate year, he is probably one of the most responsible for my being here today. For showing me during that summer that a PhD could be fun, Lee Middleton, Sylvia Wong and David Yuen must also be thanked.

I would like to thank my parents, who have provided me with the necessities, such as a home, food and Internet access, and so helped me to live while carrying out this research.

My friends have provided me with support in many ways. Toby Collett has been a friend since we began our time at Auckland University and has shared the revelations and suffering during his own PhD. Lee Middleton has played the role of LaTeX Guru and experienced senior throughout my PhD. These and other friends have been responsible for keeping me sane, particularly during the harder parts of the research.

Other members of the Robotics Group have been an invaluable sounding board for ideas, including Rick Chen, Kathy Fung, Luke Gumbley, Ben Moores, Sigrid Roehling, Grant Sargent, Dr Karl Stol, Slobodan Vukanovic, Sylvia Wong, Jack Yu, and David Yuen.

The technical and administrative staff of the department have helped ensure I had what I need when I needed it. Grant Sargent in particular has created a well-run Robotics Lab where research is fun and exciting. Lynda Jones, Bev Painter, Jamie Walker, Stephen Ward and Peter Wigan have all been very helpful.

Finally, I would like to thank The University of Auckland for providing the significant financial support I have received, and the members of the FirstOnes.com website, whose constant insanity has helped me avoid my own.

Contents

List of Figures

List of Tables

List of Listings