Lecture Notes in Artificial Intelligence 1772

Subseries of Lecture Notes in Computer Science
Edited by J. G. Carbonell and J. Siekmann

Lecture Notes in Computer Science
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo

Michael Beetz

Concurrent Reactive Plans

Anticipating and Forestalling Execution Failures

Springer

Series Editors

Jaime G. Carbonell,Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Author

Michael Beetz
Universität Bonn
Institut für Informatik III
Römerstr. 164, 53117 Bonn, Germany
E-mail: beetz@cs.uni-bonn.de

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Beetz, Michael:
Concurrent reactive plans : anticipating and forestalling execution failures / Michael Beetz. - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2000
(Lecture notes in computer science ; Vol. 1772 : Lecture notes in artificial intelligence)
ISBN 3-540-67241-9

CR Subject Classification (1991): I.2.9, I.2.11, I.2.8, I.2.6, I.2, C. 2.4, F.3, D.1.3, D.3.2

ISBN 3-540-67241-9 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by PTP Berlin, Stefan Sossna
Printed on acid-free paper SPIN 10719790 06/3142 5 4 3 2 1 0

Dedicated to the memory of my mother
Renate Beetz

and my grandparents
Elise and Fritz Lehmann

Abstract

Autonomous robots that accomplish their jobs in partly unknown and changing environments often learn important information while carrying out their jobs. To be reliable and efficient, they have to act appropriately in novel situations and respond immediately to unpredicted events. They also have to reconsider their intended course of action when it is likely to have flaws. For example, whenever a robot detects another robot, it should predict that robot's effect on its plan and — if necessary — revise its plan to make it more robust. To accomplish these patterns of activity we equip robots with *structured reactive plans (SRPs)*, concurrent control programs that can not only be interpreted but also reasoned about and manipulated. These plans specify how the robot is to respond to sensory input in order to accomplish its jobs.

In this book we describe a computational model of forestalling common flaws in autonomous robot behavior. To this end, we develop a representation for SRPs in which declarative statements for goals, perceptions, and beliefs make the structure and purpose of SRPs explicit and thereby simplify and speed up reasoning about SRPs and their projections. We have also developed a notation for transforming SRPs, which does not only make the physical effects of plan execution explicit, but also the process of plan interpretation, as well as temporal, causal, and teleological relationships among plan interpretation, the world, and the physical behavior of the robot. Using this notation a planning system can diagnose and forestall common flaws in robot plans that cannot be dealt with in other planning representations. Finally, we have extended the language for writing SRPs with constructs that allow for a flexible integration of planning and execution and thereby turned it into a single high-level language that can handle both planning and execution actions.

Experiments in a simulated world show that by simultaneously forestalling flaws and executing SRPs, the robot can perform its jobs more reliably than, and almost as efficiently as, it could using fixed control programs.

Acknowledgements

This manuscript is a revised version of my dissertation written at the Artificial Intelligence Laboratory at Yale University. In working on this dissertation, I have greatly benefited from the help and support of many people. Here are those I would like to thank especially.

First and foremost, I want to thank my advisor, Drew McDermott. He provided an excellent (intellectual and software) environment in which I could pursue my research. He was always there when I needed advice and gave me the freedom to explore my ideas. During my tenure at Yale I have benefitted immensely from his broad intuitions and his ability to turn vague ideas into working Lisp programs.

Ken Yip has been a good friend and mentor throughout. In particular, I want to thank him for teaching me that "knowing" the solution before understanding the problem is not a good way to start one's research. I hope, I won't forget too often. Greg Hager and Reid Simmons were the other two members of my Dissertation committee. Their careful reading, critique, and insightful comments have greatly improved this dissertation. I would also like to thank Pam Sturner and Zach Dodds for taking the time to read and comment on an earlier draft of this dissertation. My gratitude can best be measured by the length of this document. Sean Engelson deserves credit for being a good friend, perfect office mate, and discussion partner.

Since the completion of my thesis I have worked as a member of the RHINO team at the University of Bonn and applied techniques described in this book to problems in autonomous robot control. I would like to thank Prof. Armin B. Cremers for his support and for providing such an excellent research environment. Carrying out successful research in autonomous robot control is not possible without being a member of a team. I had the luck of joining one of the best teams: the RHINO team. I would like to thank especially the following members and alumni: Tom Arbuckle, Thorsten Belker, Wolfram Burgard, Dieter Fox, Henrik Grosskreutz, Dirk Hähnel, Dirk Schulz, Gerhard Lakemeyer, and Sebastian Thrun.

Most importantly, I want to thank my wife Annette and children Nicola and Fabian for their love and support and keeping up with me writing this manuscript.

Table of Contents

List of Figures