

Automating Acceptance Tests for Sensor- and Actuator-based Systems on the Example of Autonomous Vehicles

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der
RWTH Aachen University zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Wirtschaftsinformatiker Christian Berger

aus Braunschweig

Berichter: Universitätsprofessor Dr. rer. nat. Bernhard Rumpe
Professor Dr.-Ing. Thomas Form

Tag der mündlichen Prüfung: 19. Juli 2010

Aachener Informatik-Berichte, Software Engineering

herausgegeben von
Prof. Dr. rer. nat. Bernhard Rumpe
Software Engineering
RWTH Aachen University

Band 6

Christian Berger

**Automating Acceptance Tests for
Sensor- and Actuator-based Systems on
the Example of Autonomous Vehicles**

Shaker Verlag
Aachen 2010

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

Zugl.: D 82 (Diss. RWTH Aachen University, 2010)

Copyright Shaker Verlag 2010

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Printed in Germany.

ISBN 978-3-8322-9378-9

ISSN 1869-9170

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen

Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9

Internet: www.shaker.de • e-mail: info@shaker.de

However impenetrable it seems, if you don't try it, then you can never do it.

Sir Andrew John Wiles (British mathematician, * April 11, 1953)

Acknowledgments

This thesis was created during my work at the Software Systems Engineering Institute at Technische Universität Braunschweig, at the Department of Software Engineering at RWTH Aachen University, and at the Autonomous Ground Vehicles Group of the Center of Hybrid and Embedded Software Systems at University of California, Berkeley.

I am very grateful Prof. Dr. Bernhard Rumpe for carrying out the CarOLO project, for giving me the opportunity to realize this thesis, and for his continuous support. Furthermore, I am very grateful Prof. Dr. Thomas Form for his willingness being my second examiner and for his exceptional support for the CarOLO project. I thank Prof. Dr. Stefan Kowalewski who takes the chair of my graduation's committee and Prof. Dr. Thomas Seidl as well as Prof. Dr. Berthold Vöcking for examining me.

I would like to thank my colleagues from the Department of Software Engineering for intense and inspiring discussions, especially Ibrahim Armac, Christian Basarke, Dr. Hans Grönniger, Tim Gülke, Arne Haber, Thomas Heer, Christoph Herrmann, Dr. Anne-Thérèse Körtgen, Dr. Holger Krahn, Thomas Kurpick, Cem Mengi, Claas Pinkernell, Holger Rendel, Jan Oliver Ringert, Martin Schindler, Frank Schroven, Steven Völkel, and Ingo Weisemöller.

For many interesting insights into automotive software engineering, I thank Dr. Christian Ameling, Dr. Arne Bartels, Dr. Holger Philipps, and Dr. Dirk Stüker from Volkswagen Corporation.

Furthermore, I thank Prof. Shankar Sastry and Jan Biermeyer for the invitation to visit their Autonomous Ground Vehicles Group at University of California, Berkeley, and Humberto Gonzalez for interesting discussions about autonomous ground vehicles.

For their support for the CarOLO project I would like to thank Dr. Jan Effertz, Fred

W. Rauskolb III, and Carsten Spichalsky from Volkswagen Corporation, Prof. Dr. Peter Hecker, Prof. Dr. Marcus Magnor, Prof. Dr. Markus Lienkamp, Prof. Dr. Walter Schumacher, Prof. Dr. Ulrich Seiffert, and Prof. Dr. Lars Wolf. Furthermore, I would like to thank my former colleagues from the very exciting CarOLO project which inspired me for this thesis, especially my brother Kai Berger, Karsten Cornelsen, Michael Doering, Joop Flack, Dr. Fabian Graefe, Kai Homeier, Felix Klose, Christian Lipski, Johannes Morgenroth, Tobias Nothdurft, Sebastian Ohl, and Jörn Marten Wille. For their support during our preparations for the 2007 DARPA Urban Challenge, I would like to thank the entire team of the Southwest Research Institute in San Antonio, Texas and Tim Kuser who supported us significantly during our visit to Victorville, California.

Moreover, I thank my students Andreas Donners, Stefan Kühnel, and Jonas Schwartze supporting the realization of this work.

And last but not least, I am very grateful my parents, my family, my friends and especially my lovely wife Evelin for her continuous encouragement and patience during my entire work—this thesis is dedicated to you!

Trademarks appear throughout this thesis without any trademark symbol; they are the property of their respective trademark owner. There is no intention of infringement; the usage is to the benefit of the trademark owner.

Contents

Abstract	vii
1 Introduction and Motivation	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Main Goals and Results	4
1.4 Thesis' Structure	6
1.5 Publications	7
2 Autonomous Ground Vehicles	9
2.1 History of Autonomous Ground Vehicles	9
2.2 Architecture of Sensor- and Actuator-based Systems	14
2.2.1 Terms and Definitions	14
2.2.2 General System Architecture	15
2.3 Example: Autonomously Driving Vehicle "Caroline"	18
3 Methodology for Automating Acceptance Tests	21
3.1 General Considerations	21
3.2 Virtualizing a Sensor- and Actuator-based Autonomous System for the V-model-based Development Process	22
3.2.1 Preconditions and Limitations	25
3.2.2 Formal Specification of the System's Context	26
3.2.3 Specification of Customer's Acceptance Criteria	27
3.2.4 Decoupling the SUD's Evaluation from the Real Hardware En- vironment and System's Context	29

3.2.5	Structure for the Following Chapters	33
4	Modeling the System’s Context	35
4.1	General Considerations and Design Drivers	35
4.2	Mathematical Considerations	36
4.2.1	Manipulations in \mathbb{R}^3	37
4.2.2	Quaternions	39
4.2.3	The WGS84 Coordinate System	42
4.3	Domain Analysis: Surroundings of Autonomous Ground Vehicles	43
4.3.1	Traits of Private Areas	44
4.3.2	Traits of Public Areas	44
4.3.3	Surroundings’ Elements	45
4.4	Modeling the Surroundings of Autonomous Ground Vehicles	46
4.4.1	Modeling of Stationary Elements	47
4.4.2	Modeling of Dynamic Elements	50
4.5	Scenario-based Modeling	51
4.5.1	Scenario-based Modeling Using MontiCore	52
4.5.2	Scenario-based Modeling Using C++	54
5	The Framework <i>Hesperia</i>	61
5.1	General Considerations and Design Drivers	61
5.2	Software Architecture	63
5.3	Encapsulating the Operating System and Required Third Party Li- braries: <code>libcore</code>	66
5.3.1	Package wrapper	66
5.3.2	Basic Concepts	70
5.4	Providing Extensible and Re-Usable High-Level Concepts: <code>libhesperia</code>	79
5.4.1	Concept: <i>ClientConference</i>	79
5.4.2	Concept: <i>Dynamic Module Configuration</i>	79
5.4.3	Concept: Enhanced Data Structures	81
5.4.4	Concept: Integration of Modeling DSL	84
5.4.5	Concept: Device-Independent Data Visualization	85

5.5	Applications' Life Cycle Management: <code>supercomponent</code>	87
5.6	Supporting Components	87
5.6.1	Component: <code>proxy</code>	88
5.6.2	Component: <code>recorder</code>	88
5.6.3	Component: <code>player</code>	88
5.6.4	Component: <code>rec2video</code>	89
6	Simulation of the System's Context	91
6.1	General Considerations and Design Drivers	91
6.2	Controlling an SUD and the Time	94
6.3	Controlling Applications: <code>libcontext</code>	98
6.3.1	Interface for Controlling Time and Communication	99
6.3.2	Supervising an Application's Control Flow and Communication	101
6.3.3	Remarks	110
6.4	Providing an SUD-dependent System Context: <code>libvehiclecontext</code>	112
6.4.1	Unattended System Simulations using <code>libvehiclecontext</code>	112
6.4.2	Interactive System Simulations using <code>libvehiclecontext</code>	113
6.4.3	Position Provider: Bicycle Model	114
6.4.4	Monocular Camera Provider	119
6.4.5	Stereo Camera Provider	120
6.4.6	Single Layer Laser Scanner Provider	121
6.4.7	Sensor-fusion Provider	128
6.4.8	Dynamic Context Provider	132
7	Monitoring and Unattended Reporting	135
7.1	General Considerations and Design Drivers	135
7.2	Interactive Monitoring	136
7.2.1	Monitoring Component <code>monitor</code>	136
7.2.2	Visualization of Stationary Surroundings	139
7.2.3	Visualization of Dynamic Elements	141
7.3	Unattended and Automatic Monitoring of an SUD for Acceptance Tests	143
7.3.1	Architecture for the Reporting Interface	143
7.3.2	Usage of Reporting Interface	145

8	Case Study and Evaluation	155
8.1	Benchmark for <i>Hesperia</i>	155
8.1.1	Performance of the Timing	155
8.1.2	Performance of the Communication	158
8.2	Application for <i>Hesperia</i> on an Autonomous Ground Vehicle	163
8.2.1	Ford Escape Hybrid – ByWire XGV	164
8.2.2	Test Site “Richmond Field Station”	168
8.2.3	Modeling the Ford Escape Hybrid XGV	169
8.2.4	Velocity and Steering Control Algorithm	173
8.2.5	Automating Test Runs	186
8.3	Further Applications of the Framework <i>Hesperia</i>	189
8.3.1	Sensor Data Collection	189
8.3.2	Virtual Sensor Data Collection	190
8.3.3	Application-Dependent Additive Sensor Data Generation	190
8.3.4	Evaluation Runs	190
8.3.5	Illustrating a Sensor- and Actuator-based Autonomous System’s Performance	192
9	Related Work	193
9.1	Frameworks for Distributed Component-Based Embedded Automotive Software	193
9.1.1	Elektrobit Automotive GmbH: Automotive Data and Time Triggered Framework	194
9.1.2	AUTOSAR	195
9.1.3	OpenJAUS	196
9.1.4	Orca/Hydro	198
9.1.5	Evaluation of the Frameworks for Distributed Component-Based Embedded Automotive Software	199
9.1.6	Other Robotics and Communication Frameworks	199
9.2	Software Development and System Testing	202
9.2.1	Driving Simulator at Deutsches Zentrum für Luft und Raumfahrt (DLR)	202

9.2.2	Framework for Simulation of Surrounding Vehicles in Driving Simulators	203
9.2.3	The Iowa Driving Simulator	203
9.2.4	IPG Automotive GmbH: CarMaker	205
9.2.5	Microsoft Corporation: Robotics Developer Studio	206
9.2.6	PELOPS	206
9.2.7	Player/Stage/Gazebo	207
9.2.8	TESIS Gesellschaft für Technische Simulation und Software: DYNAware	208
9.2.9	IAV GmbH: Test Environment for Synthetic Environment Data .	208
9.2.10	TNO PreScan	209
9.2.11	VIRES Simulationstechnologie GmbH: Virtual Test Drive . . .	210
9.2.12	Evaluation of Approaches for Software Development and System Testing	211
10	Conclusion And Future Work	213
	Bibliography	219
	List of Figures	237
	List of Equations	252
	List of Listings	253
A	Grammar for Defining the System's Context	257
A.1	MontiCore Grammar for the Stationary Surroundings	257
A.2	MontiCore Grammar for the Dynamic Surroundings	265
B	List of Abbreviations	271
C	Thesis' Structure	275
D	Curriculum Vitae	279

Abstract

In projects dealing with autonomous vehicles which are driving in different contexts like highways, urban environments, and rough areas, managing the software's quality for the entire data processing chain of sensor- and actuator-based autonomous systems is increasingly complex. One main reason is the early dependency on all sensor's raw data to setup the data processing chain and to identify subsystems. These sensors' data might be extensive, especially if laser scanners or color camera systems are continuously producing a vast amount of raw data. Moreover, due to this dependency the sensors' setup including their respectively specified mounting positions and calibration information is also necessary to gather real input data from real surroundings' situations of the system. This is even more important before actually starting to integrate independently developed subsystems for carrying out tests for the entire data processing chain.

To reduce this dependency and therefore to decouple tasks from the project's critical path, an approach is outlined in this thesis which was developed to support the software engineering for sensor- and actuator-based autonomous systems. This approach relies on customer's requirements and corresponding customer's acceptance criteria as well as the decoupling of the software engineering from the real hardware environment to allow appropriate system simulations.

Based on the customer's requirements, formally specified scenarios using a domain specific language are derived which are used to provide surroundings and suitable situations for a sensor- and actuator-based autonomous system. From these formally specified surroundings, the required input data is derived for different layers of a sensor data processing system to generate actions within the system's context. This input data itself depends on a given sensor model to compute its specific raw data. Amongst others,

on the example of laser scanners and camera systems, algorithms using modern graphical processing units are outlined to generate the required raw data even for complex situations.

To realize the aforementioned aspects, a development environment is necessary consisting of tools for modeling and working with instances of a domain specific language. Furthermore, a software framework is required which provides easily usable and mature solutions for common programming requirements like synchronization for concurrent threads or communication in a high-level point of view. For relying on a consistent and homogeneous software framework for implementing the concepts, a highly portable and real-time-capable software framework for distributed applications was realized which was written entirely from scratch in strictly object-oriented C++. Moreover, this software framework also integrates the formally modeled input data derived from the specified requirements and the sensors' models to allow unattended system simulations to support the acceptance tests for subsystems or an entire system.

On the example of autonomous vehicles, the applicability of the approach and the software framework is demonstrated by implementing a vehicle navigation algorithm which uses a given digital map for finding the best route to a desired destination from an arbitrarily chosen starting point. This algorithm was developed considering the test-first-principle and is continuously evaluated by unattended and automatic software tests which are executed on a continuous integration system. Its implementation as well as its evaluation make use of the aforementioned concepts and algorithms. Therefore, the vehicle's surroundings were formally modeled together with its necessary sensors using the provided development tools and environment for performing and evaluating unattended system runs before the algorithm was put into operation on the real vehicle.