Lecture Notes in Mechanical Engineering

Jürgen Gausemeier
Franz Josef Rammig
Wilhelm Schäfer
Walter Sextro *Editors*

# Dependability of Self-optimizing Mechatronic Systems

Springer

Jürgen Gausemeier · Franz Josef Rammig
Wilhelm Schäfer · Walter Sextro
Editors

# Dependability of Self-optimizing Mechatronic Systems

Springer

*Editors*
Jürgen Gausemeier
Product Engineering
Heinz Nixdorf Institute
University of Paderborn
Paderborn
Germany

Wilhelm Schäfer
Software Engineering
Heinz Nixdorf Institute
University of Paderborn
Paderborn
Germany

Franz Josef Rammig
Design of Distributed Embedded Systems
Heinz Nixdorf Institute
University of Paderborn
Paderborn
Germany

Walter Sextro
Chair of Mechatronics and Dynamics
University of Paderborn
Paderborn
Germany

ISSN 2195-4356 ISSN 2195-4364 (electronic)
ISBN 978-3-642-53741-7 ISBN 978-3-642-53742-4 (eBook)
DOI 10.1007/978-3-642-53742-4
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013956528



Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Lecture Notes in Mechanical Engineering

For further volumes:
http://www.springer.com/series/11236

*About this Series*

Lecture Notes in Mechanical Engineering (LNME) publishes the latest developments in Mechanical Engineering - quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNME. Also considered for publication are monographs, contributed volumes and lecture notes of exceptionally high quality and interest. Volumes published in LNME embrace all aspects, subfields and new challenges of mechanical engineering. Topics in the series include:

- Engineering Design
- Machinery and Machine Elements
- Mechanical Structures and Stress Analysis
- Automotive Engineering
- Engine Technology
- Aerospace Technology and Astronautics
- Nanotechnology and Microengineering
- Control, Robotics, Mechatronics
- MEMS
- Theoretical and Applied Mechanics
- Dynamical Systems, Control
- Fluid Mechanics
- Engineering Thermodynamics, Heat and Mass Transfer
- Manufacturing
- Precision Engineering, Instrumentation, Measurement
- Materials Engineering
- Tribology and Surface Technology

# Preface

Intelligent technical systems offer the tempting possibility to create products that are far more capable than today's mechatronic systems. However, the system-inherent intelligence comes with increased system complexity. This leads to an increased risk for system-inherent faults, which might have devastating effects such as a complete breakdown or might even compromise the safety of users. Yet at the same time the new capabilities can be used to actively compensate for faults. Among these intelligent technical systems are self-optimizing systems, which are able to adapt their behavior based on user demands, the current situation or the system state itself.

Within the Collaborative Research Center (CRC) 614 "Self-Optimizing Concepts and Structures in Mechanical Engineering" at the University of Paderborn, we devoted our research to the development of these self-optimizing systems. They are based on mechatronics, which is formed by close collaboration of mechanical, electrical and software engineering. To make a system intelligent, the software engineering aspects gain further significance and are enhanced by advanced control engineering and mathematical methods. One result of our work is the accompanying book "Design Methodology for Intelligent Technical Systems", which introduces our proposed development process for self-optimizing systems. This process needs be accompanied by dedicated methods to assure or increase system dependability as early as possible in order to avoid costly design changes. We found that this augmenting process is so important yet also challenging that we decided to separate all related works into this book. It was written by the members of the CRC 614's working group "Safety and Stability" and coordinated by Tobias Meyer, to whom I'd like to express my sincerest gratitude.

The results of the CRC 614 and the book at hand present a major milestone towards the development of dependable intelligent systems. We hope that it inspires you to create tomorrow's products!

Your CRC-Team

Paderborn,
October 2013

Prof. Dr.-Ing. Jürgen Gausemeier
Speaker of the Collaborative Research Center 614

# Acknowledgements

This book was written within the context of the Collaborative Research Center 614 "Self-Optimizing Concepts and Structures in Mechanical Engineering", which has been funded by the Deutsche Forschungsgemeinschaft (DFG).

# List of Contributors

Prof. Dr.-Ing. habil. Wilhelm Dangelmaier
Heinz Nixdorf Institute, University of Paderborn, Business Computing, especially CIM, Fuerstenallee 11, D-33102 Paderborn, Germany,
e-mail: wilhelm.dangelmaier@hni.uni-paderborn.de

Prof. Dr. Michael Dellnitz
Chair of Applied Mathematics, University of Paderborn, Warburger Straße 100, 33098 Paderborn, Germany,
e-mail: dellnitz@uni-paderborn.de

Rafal Dorociak
Heinz Nixdorf Institute, University of Paderborn, Product Engineering, Fuerstenallee 11, D-33102 Paderborn, Germany,
e-mail: rafal.dorociak@hni.uni-paderborn.de

Kathrin Flaßkamp
Computational Dynamics and Optimal Control, University of Paderborn, Warburger Straße 100, 33098 Paderborn, Germany,
e-mail: kathrinf@math.uni-paderborn.de

Prof. Dr.-Ing. Juergen Gausemeier
Heinz Nixdorf Institute, University of Paderborn, Product Engineering, Fuerstenallee 11, D-33102 Paderborn, Germany,
e-mail: juergen.gausemeier@hni.uni-paderborn.de

Stefan Groesbrink
Heinz Nixdorf Institute, University of Paderborn, Design of Distributed Embedded Systems, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: stefan.groesbrink@hni.uni-paderborn.de

Philip Hartmann
Heinz Nixdorf Institute, University of Paderborn, Business Computing, especially CIM, Fuerstenallee 11, D-33102 Paderborn, Germany,
e-mail: philip.hartmann@hni.uni-paderborn.de

Christian Heinzemann
Heinz Nixdorf Institute, University of Paderborn, Software Engineering Group, Zukunftsmeile 1, 33102 Paderborn, Germany,
e-mail: c.heinzemann@uni-paderborn.de

Christian Hölscher
Design and Drive Technology, University of Paderborn, Pohlweg 47-49, 33098 Paderborn, Germany,
e-mail: c.hoelscher@uni-paderborn.de

Peter Iwanek
Heinz Nixdorf Institute, University of Paderborn, Product Engineering, Fuerstenallee 11, D-33102 Paderborn, Germany,
e-mail: peter.iwanek@hni.uni-paderborn.de

Jan Henning Keßler
Heinz Nixdorf Institute, University of Paderborn, Control Engineering and Mechatronics, Fuerstenallee 11, D-33102 Paderborn, Germany,
e-mail: jan.henning.kessler@hni.uni-paderborn.de

Dr. Bernd Kleinjohann
Heinz Nixdorf Institute, University of Paderborn, Design of Distributed Embedded Systems, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: bernd@c-lab.de

Dr. Lisa Kleinjohann
Heinz Nixdorf Institute, University of Paderborn, Design of Distributed Embedded Systems, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: lisa@c-lab.de

Sebastian Korf
Heinz Nixdorf Institute, University of Paderborn, System and Circuit Technology, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: korf@hni.uni-paderborn.de

Martin Krüger
Heinz Nixdorf Institute, University of Paderborn, Control Engineering and Mechatronics, Fuerstenallee 11, D-33102 Paderborn, Germany,
e-mail: kruemar@uni-paderborn.de

Tobias Meyer
Mechatronics and Dynamics, University of Paderborn, Pohlweg 47-49, 33098 Paderborn, Germany,
e-mail: tobias.meyer@uni-paderborn.de

Dr. Wolfgang Müller
Heinz Nixdorf Institute, University of Paderborn, Design of Distributed Embedded Systems, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: wolfgang@c-lab.de

Jun.-Prof. Dr. Sina Ober-Blöbaum
Computational Dynamics and Optimal Control, University of Paderborn, Warburger Straße 100, 33098 Paderborn, Germany,
e-mail: sinaob@math.uni-paderborn.de

Dr.-Ing. Mario Porrmann
Cognitronics and Sensor Systems, Center of Excellence Cognitive Interaction Technology, Bielefeld University,Universitätsstraße 21-23, 33615 Bielefeld, Germany,
e-mail: mporrmann@cit-ec.uni-bielefeld.de

Claudia Priesterjahn
Heinz Nixdorf Institute, University of Paderborn, Software Engineering Group, Zukunftsmeile 1, 33102 Paderborn, Germany,
e-mail: c.priesterjahn@uni-paderborn.de

Prof. Dr. Franz-Josef Rammig
Heinz Nixdorf Institute, University Paderborn, Design of Distributed Embedded Systems, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: franz@uni-paderborn.de

Christoph Rasche
Heinz Nixdorf Institute, University of Paderborn, Design of Distributed Embedded Systems, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: crasche@c-lab.de

Peter Reinold
Heinz Nixdorf Institute, University of Paderborn, Control Engineering and Mechatronics, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: peter.reinold@hni.upb.de

Prof. Dr. Wilhelm Schäfer
Heinz Nixdorf Institute, University of Paderborn, Software Engineering Group, Zukunftsmeile 1, 33102 Paderborn, Germany,
e-mail: wilhelm@uni-paderborn.de

Albert Seifried
University of Paderborn, Warburger Straße 100, 33098 Paderborn, Germany,
e-mail: seifried@math.uni-paderborn.de

Prof. Dr.-Ing. habil Walter Sextro
Mechatronics and Dynamics, University of Paderborn, Pohlweg 47-49, 33098 Paderborn, Germany,
e-mail: walter.sextro@uni-paderborn.de

Christoph Sondermann-Woelke
Mechatronics and Dynamics, University of Paderborn, Pohlweg 47-49, 33098 Paderborn, Germany,
e-mail: christoph.sondermann-woelke@uni-paderborn.de

Katharina Stahl
Heinz Nixdorf Institute, University of Paderborn, Design of Distributed Embedded Systems, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: katharina.stahl@uni-paderborn.de

Dominik Steenken
Specification and Modelling of Software Systems, University of Paderborn, Warburger Straße 100, 33098 Paderborn, Germany,
e-mail: dominik@uni-paderborn.de

Robert Timmermann
Applied Mathematics, University of Paderborn, Warburger Straße 100, 33098 Paderborn, Germany,
e-mail: robert.timmermann@math.upb.de

Prof. Dr.-Ing. habil. Ansgar Trächtler
Heinz Nixdorf Institute, University of Paderborn, Control Engineering and Mechatronics, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: ansgar.traechtler@hni.upb.de

Mareen Vaßholz
Heinz Nixdorf Institute, University of Paderborn, Product Engineering, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: mareen.vassholz@hni.uni-paderborn.de

Prof. Dr. Heike Wehrheim
Specification and Modelling of Software Systems, University of Paderborn, Warburger Straße 100, 33098 Paderborn, Germany,
e-mail: wehrheim@uni-paderborn.de

Dr. Katrin Witting
Applied Mathematics, University of Paderborn, Warburger Straße 100, 33098 Paderborn, Germany,
e-mail: witting@math.upb.de

Tao Xie
Heinz Nixdorf Institute, University of Paderborn, Design of Distributed Embedded Systems, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: tao@c-lab.de

Yuhong Zhao
Heinz Nixdorf Institute, University of Paderborn, Design of Distributed Embedded Systems, Fuerstenallee 11, 33102 Paderborn, Germany,
e-mail: zhao@uni-paderborn.de

Steffen Ziegert
Specification and Modelling of Software Systems, University of Paderborn, Warburger Straße 100, 33098 Paderborn, Germany,
e-mail: `steffen.ziegert@uni-paderborn.de`

Prof. Dr.-Ing. Detmar Zimmer
Design and Drive Technology, University of Paderborn, Pohlweg 47-49, 33098 Paderborn, Germany,
e-mail: `detmar.zimmer@uni-paderborn.de`

# Contents

# Chapter 1
# Introduction to Self-optimization and Dependability

**Abstract.** This chapter gives an introduction to self-optimizing mechatronic systems and the risks and possibilities that arise with these. Self-optimizing mechatronic systems have capabilities that go far beyond those of traditional mechatronic systems. They are able to autonomously adapt their behavior and so react to outer influences, which can originate e.g. from the environment, changed user requirements or the current system status. The basic process of self-optimization, the procedures employed within and the main components of a self-optimizing system are explained here.

However, during the development of such a system, several challenges need to be met. On this note, both the concept of dependability and our proposed development process for self-optimizing systems are introduced. This process is used to derive a methodology for the selection of dependability methods in the development of self-optimizing systems.

To illustrate the proposed process and methods, several demonstrators are introduced. These are self-optimizing systems from different fields of engineering, e.g. robotics, automotive engineering and railroad engineering.

The chapter concludes with an overview of the content of the remainder of the book.

Tobias Meyer, Claudia Priesterjahn, and Walter Sextro

In today's market, a growing number of competitive and successful products are mechatronic systems. Mechatronic systems provide an advanced functionality that is achieved by a collaboration between mechanical engineering, electrical engineering, control engineering, software engineering, and mathematics. The combination of these domains allows the implementation of self-optimizing capabilities, which enable a system to react autonomously to changes in the system requirements or the environment. Moreover, high dependability of these systems is an absolute must, as self-optimizing mechatronic systems are often used in environments in which safety is critical. However, due to the complexity of self-optimizing mechatronic systems, ensuring dependability becomes a difficult task.

J. Gausemeier et al. (eds.), *Dependability of Self-optimizing Mechatronic Systems*,
Lecture Notes in Mechanical Engineering,
DOI: 10.1007/978-3-642-53742-4_1, 

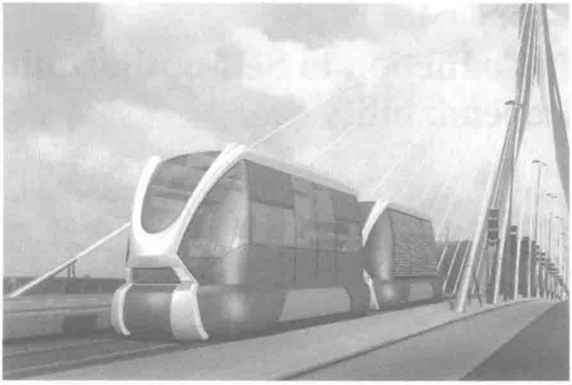

**Fig. 1.1** Two RailCabs forming a convoy (artist's interpretation)

One practical example of a self-optimizing mechatronic system is the RailCab system, shown in Fig. 1.1. The vision of the RailCab project is a mechatronic rail system where small autonomous vehicles, called RailCabs, provide flexible transport on rail tracks. RailCabs form convoys without being rigidly coupled, in order to reduce energy consumption caused by air resistance. Such convoys are established on demand, and require only small distances between the RailCabs. The small distances required make the RailCabs a safety-critical system requiring high dependability.

No prior approach to the development of self-optimizing systems has taken the inherent risks and capabilities of these systems into account. The development process, which is presented in the book "Design Methodology for Intelligent Technical Systems" [14], coordinates the different domains that are involved in the development of a mechatronic system. However, this process needs to be complemented by methods enabling the developer to guarantee the dependability of self-optimizing mechatronic systems.

In order to overcome the apparent contradiction between the additional complexity introduced by self-optimization and the high dependability requirements, self-optimization capabilites within the system itself can be employed to increase the dependability of a system. This can be achieved by using behavior adaptation based on the current system state. In this book, we present methods for the development of self-optimizing mechatronic systems that either focus on ensuring the dependability of such systems or that use the capability of self-optimizing systems to increase their dependability system-internally. These methods are integrated into the development process for self-optimizing mechatronic systems in the book "Design Methodology for Intelligent Technical Systems" [14] and thus allow for early reaction to possible risks.

Using the development process presented in this book, the RailCab has become an example of a dependable self-optimizing mechatronic system.

## 1.1 Self-optimizing Mechatronic Systems

Michael Dellnitz, Kathrin Flaßkamp, Philip Hartmann, Martin Krüger, Tobias Meyer, Claudia Priesterjahn, Sina Ober-Blöbaum, Christoph Rasche, Walter Sextro, Katharina Stahl, and Ansgar Trächtler

Self-optimizing systems are intelligent mechatronic systems that are able to adapt their behavior autonomously to changing operating conditions and user demands. To this end, three steps are continuously cycled, as discussed in [12, 13], see also Fig. 1.2:

1. Analysis of current situation:
   During this step, the user requirements, the system state and based on direct observations the environment are identified and considered. Observations can also be indirectly obtained by communicating with other systems, and the state of the system may include prior observations. A major aspect of this step is the evaluation in how far the given objectives have been fulfilled.
2. Determination of objectives:
   In the second step, the new system objectives can be selected, adapted or generated from the system's previous objectives. Selection is possible if a finite number of discrete possible objectives exist. Adaptation is carried out if the system's objectives can be altered gradually. The generation of new objectives is performed if new objectives must be created independently of known objectives.
3. Adaptation of system behavior:
   The system behavior is adapted to account for changes necessitated by the determination of objectives. Changes to system parameters as well as changes to the system structure are possible. This action constitutes the feedback of the self-optimization cycle to the system.

As already mentioned, the objectives are of major importance for a self-optimizing system. They are organized in the System of Objectives, which in general is a set

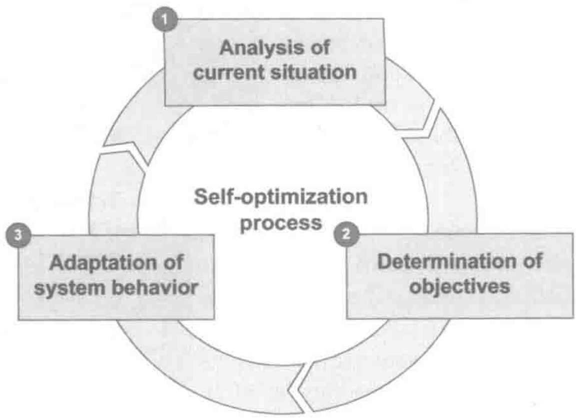


**Fig. 1.2** Cycle of the behavior adaptation of a self-optimizing mechatronic system