FORMAL TECHNIQUES IN REAL-TIME AND FAULT-TOLERANT SYSTEMS

THE KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE

REAL-TIME SYSTEMS

Consulting Editor

John A. Stankovic

REAL-TIME UNIX SYSTEMS: *Design and Application Guide,*
B. Furht, D. Grostick, D. Gluch, G. Rabbat, J. Parker, M. McRoberts,
ISBN: 0-7923-9099-7

FOUNDATIONS OF REAL-TIME COMPUTING: *Scheduling and Resource Management,* A. M. van Tilborg, G. M. Koob; ISBN: 0-7923-9166-7

FOUNDATIONS OF REAL-TIME COMPUTING: *Formal Specifications and Methods,* A. M. van Tilborg, G. M. Koob; ISBN: 0-7923-9167-5

CONSTRUCTING PREDICTABLE REAL TIME SYSTEMS,
W. A. Halang, A. D. Stoyenko; ISBN: 0-7923-9202-7

SYNCHRONIZATION IN REAL-TIME SYSTEMS: *A Priority Inheritance Approach,* R. Rajkumar; ISBN: 0-7923-9211-6

REAL-TIME SYSTEMS ENGINEERING AND APPLICATIONS,
M. Schiebe, S. Pferrer; ISBN: 0-7923-9196-9

SYNCHRONOUS PROGRAMMING OF REACTIVE SYSTEMS,
N. Halbwachs; ISBN: 0-7923-9311-2

FORMAL TECHNIQUES IN REAL-TIME AND FAULT-TOLERANT SYSTEMS

edited
by

Jan Vytopil
BSO/Origin
University of Nijmegen

SPRINGER SCIENCE+BUSINESS MEDIA, LLC

Library of Congress Cataloging-in-Publication Data

Formal techniques in real-time and fault-tolerant systems / edited by Jan Vytopil.
p. cm. -- (Kluwer international series in engineering and computer science ; 221. Real-time systems)
Includes bibliographical references and index.
ISBN 978-1-4613-6414-6 ISBN 978-1-4615-3220-0 (eBook)
DOI 10.1007/978-1-4615-3220-0
1. Real-time data processing. 2. Fault-tolerant computing. I. Vytopil, J. (Jan), 1947- . II. Series: Kluwer international series in engineering and computer science ; SECS 221. III. Series: Kluwer international series in engineering and computer science. Real-time systems.
QA76.54.F65 1993
004'.33--dc20 93-16676
CIP

CONTENTS

CONTRIBUTORS vii

PREFACE ix

I CONCEPTS AND FOUNDATIONS 1

1 *Henk Schepers*
Terminology and Paradigms for Fault Tolerance 3

2 *Doug G. Weber*
Fault Tolerance as Self-Similarity 33

3 *Jos Coenen and Jozef Hooman*
Parameterized Semantics for Fault Tolerant Real-Time Systems 51

4 *Hans A. Hansson*
Modeling Real-Time and Reliability 79

II APPLICATIONS 107

5 *John Rushby*
A Fault-Masking and Transient-Recovery Model
for Digital Flight-Control Systems 109

6 *Zhiming Liu and Mathai Joseph*
Specification and Verification of Recovery in
Asynchronous Communicating Systems 137

7 *Jan Peleska*
CSP, Formal Software Engineering and the Development of
Fault-Tolerant Systems 167

INDEX 207

CONTRIBUTORS

Jos Coenen
Department of Mathematics
and Computing Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven
The Netherlands
E-mail: wsinjosc@win.tue.nl

Hans A. Hansson
Swedish Institute of Computer Science
Box 1263, S-164 28 Kista
and
Department of Computer Science
Uppsala University
Sweden
E-mail: hansh@sics.se

Jozef Hooman
Department of Mathematics
and Computing Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven
The Netherlands
E-mail: wsinjh@win.tue.nl

Mathai Joseph
Department of Computer Science
University of Warwick
Coventry CV4 7AL, Warwick
United Kingdom
E-mail: mathai@dcs.warwick.ac.uk

Zhiming Liu
Department of Computer Science
University of Warwick
Coventry CV4 7AL, Warwick
United Kingdom
E-mail: liu@dcs.warwick.ac.uk

Jan Peleska
Deutsche System–Technik GmbH
Edisonstraße 3, D-2300 Kiel 14
Federal Republic of Germany
E-mail: jap@informatik.uni-kiel.dbp.de

John Rushby
Computer Science Laboratory
SRI International
Menlo Park CA 94025
United States of America
E-mail: rushby@csl.sri.com

Henk Schepers
Department of Mathematics
and Computing Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven
The Netherlands
E-mail: schepers@win.tue.nl

Doug G. Weber
118 West Enfield Center Road
Ithaca, NY 14850
United States of America
E-mail: weber@keysoft.com

PREFACE

Practically every day, the media report that malfunctioning of a computer system resulted in incidents. This does not necessarily mean that the software and hardware making up such a system has not been designed with as much care as is commercially feasible. However, as the burden of controlling complicated systems is shifted onto computers, so does the complexity of computer software and hardware increase.

The sobering description of failures of some systems has led to the belief that there is a need for a distinct engineering discipline with its own theoretical foundations, objective design standards and supporting tools in order to develop reliable systems.

The term 'reliability (of a system or its components)' in computer science is often defined as the "probability that a certain system component functions correctly over a certain period of time". This requires that reliability is modelled in a time-dependant, quantitative probabilistic formal framework. However, reasoning about correctness of a system — i.e. an ability to deliver an a priori defined function, which is a qualitative issue — can be separated from quantitative probabilistic notions of reliability. A reliability of a system (or a subsystem) in qualitative sense can be expressed in terms of properties that qualitatively characterize the behaviour of a system that is error-prone.

The term 'fault-tolerance' describes that a system has properties which enable it to deliver its specified function despite of (certain) faults of its subsystem. Fault-tolerance is achieved by adding extra hardware and/or software which corrects the effects of faults. In this sense, a system can be called fault-tolerant if it can be proved the resulting (extended) system under some model of reliability meets the reliability requirements.

The chapters in this volume deal mostly with reliability from a qualitative point of view. It contains a selection of papers that focus on the state-of-the-art in formal specification, development and verification of fault-tolerant computing systems. Preliminary versions of some papers were presented at the School and

Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems held at University of Nijmegen in January 1992. Other chapters are written versions of lectures and tutorials presented at the same event.

The main theme of this volume can be formulated as follows: How does a specification, development and verification of conventional and fault- tolerant systems differ? How do the notations, methodology and tools used in design and development of fault-tolerant and conventional systems differ?

The purpose of this book is to explore these important issues, the definite answers, if they exist at all, are in my opinion still some years in the future.

The book is divided in two parts: **Concepts and Foundations** and **Applications**. Each part contains a number of contributions written by different researchers. Each chapter is self-contained and may be profitably studied without prior detailed familiarity with previous chapters. However, it is advisable to examine each chapter carefully because only then do many of the important and subtle differences in approach become evident.

The First Part: **Concepts and Foundations** sets the stage for what follows by defining the basic notions and practices of the field of design and specification of fault-tolerant systems. The chapter by Henk Schepers: "Terminology and Paradigms for Fault Tolerance" analyses the interaction between fault-hypothesis and design decisions.

A definition of the notion "fault-tolerance" that does not refer, as usually, to the functional correctness properties is given in chapter "Fault-Tolerance as Self-Similarity" by Doug Weber.

The chapter "Parameterized Semantics for Fault Tolerant Real-Time Systems" by Jos Coenen and Jozef Hooman presents a denotational semantics to describe real-time behaviour of distributed programs. In this semantics, the occurrences of hardware faults and their effects on real-time behaviour of programs can be modelled.

The effects of these faults upon the behaviour of the programs can be described as well. Hans A. Hansson in his chapter "Modeling Real-time and Reliability" provides a framework for specification and verification of distributed systems in which the reliability, timeliness and functionality can be modelled.

The Second Part: **Applications** is the "how-to" Part. It contains examples of the use of formal methods in specification and development of fault-tolerant sys-

tems. The chapter by John Rushby: "A Fault-Masking and Transient-Recovery Model for Digital Flight-Control Systems" presents a formal model and analysis for fault-masking and transient-recovery among replicated computers of digital flight-control system. This model has been specified in the language of EHDM and the crucial theorem and its corollary have been mechanically checked.

Zhiming Liu and Mathai Joseph in their chapter "Specification and Verification of Recovery in Asynchronous Communicating Systems" presents a method for specification and verification of general checkpointing programs. It combines the considerations of checkpointing, interference by physical faults and subsequent recovery so that the properties of fault-tolerant programs can be proved.

The chapter by Jan Peleska, "CSP, Formal Software Engineering and the Development of Fault-tolerant Systems", describes the use of formal techniques in development of flight control system in real industrial environment. In this article the Structured Method of Ward and Mellor is combined with formal specification language CSP of C.A.R. Hoare. The transformation schemata of Ward and Mellor are interpreted by means of translation rules so that a structured specification can be transformed into a CSP program. The use of the method is illustrated by showing that a dual computer system is tolerant to certain types of failures.

This book is suitable for graduate or advanced undergraduate course use when supplemented by additional readings that place the material contained herein in fuller context. Most of the techniques and notations described in this book are not yet ready for widespread use in commercial settings although some have been used in realistic setting.

FORMAL TECHNIQUES IN REAL-TIME AND FAULT-TOLERANT SYSTEMS