



Carnegie Mellon University
Software Engineering Institute

A Practitioner's Handbook for Real-Time Analysis

**Guide to Rate Monotonic
Analysis for Real-Time Systems**

**Mark H. Klein
Thomas Ralya
Bill Pollak
Ray Obenza**

Software Engineering Institute

Michael González Harbour
Universidad de Cantabria, Spain



Springer Science+Business Media, LLC

Library of Congress Cataloging-in-Publication Data

A Practitioner's handbook for real-time analysis : guide to rate monotonic analysis for real-time systems / Mark Klein ... [et al.].

p. cm. -- (The Kluwer international series in engineering and computer science. Real-time systems)

At head of title: Carnegie-Mellon University, Software Engineering Institute.

Includes bibliographical references and index.

ISBN 978-1-4613-6209-8 ISBN 978-1-4615-2796-1 (eBook)

DOI 10.1007/978-1-4615-2796-1

1. Real-time data processing. 2. System analysis. I. Klein, Mark. II. Carnegie-Mellon University. Software Engineering Institute. III. Series.

QA76.54.P73 1993

005.2--dc20

93-10744

CIP

Copyright © 1993 by Springer Science+Business Media New York
Originally published by Kluwer Academic Publishers in 1993
Softcover reprint of the hardcover 1st edition 1993

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, mechanical, photo-copying, recording, or otherwise, without the prior written permission of the publisher. Springer Science+Business Media, LLC.

Printed on acid-free paper.

This printing is a digital duplication of the original edition.

**THE KLUWER INTERNATIONAL SERIES
IN ENGINEERING AND COMPUTER SCIENCE**

**The Software Engineering Institute
Real-Time Handbook Series**



Table of Contents

Preface P-1
Acknowledgments Ack-1

Part 1 Introduction

Chapter 1 About This Handbook 1-1
 How Is this Book Organized? 1-2
 How to Use this Book 1-7

Chapter 2 Fundamentals of RMA 2-1
 Periodic Tasks 2-2
 Task Interaction 2-4
 Aperiodic Tasks 2-8
 Key Principles of RMA 2-15

Part 2 Concepts and Techniques

Chapter 3 A Framework for Describing Real-Time Systems 3-1
 Introduction to the Framework 3-2
 Description of the Framework 3-5
 Syntax for Describing Real-Time Situations 3-23
 Representing Real-Time Situations 3-34
 Using the Framework to Find Your Situation 3-43
 Decision Tables 3-47

Chapter 4 Techniques for Analyzing Timing Behavior 4-1
 Getting a Sense for Schedulability 4-9
 Getting a Precise Schedulability Assessment 4-19
 Assessing Capacity 4-54



Part 3 Analyzing Real-Time Systems

Chapter 5 Basic Real-Time Situations 5-1

Handling Periodic Events 5-2

Designing Tasks that Must Synchronize to Share Common Data 5-25

Handling Aperiodic Events 5-61

Chapter 6 Advanced Real-Time Situations 6-1

Controlling Jitter 6-2

Message Passing Paradigms 6-64

Issues in Multiprocessor and Distributed Systems 6-94

Chapter 7 Effects of Operating System and Runtime Services on Timing Analysis 7-1

New Events and Actions Introduced by Operating System Services 7-5

Limited Representations 7-28

Presence or Absence of Particular Operating System Features 7-38

Support for RMA Concepts in Standard Operating and Runtime Systems 7-60

Part 4 Using the Handbook on Realistic Systems

Chapter 8 Analyzing Complex Systems 8-1

Guidelines for Applying Rate Monotonic Analysis 8-2

Message Passing (MP) System Description 8-3

Applying RMA Guidelines to the Message Passing System 8-5

Chapter 9 Designing with Rate Monotonic Analysis 9-1

Guidelines for Applying Rate Monotonic Analysis 9-2

Description of Sonar Transmission Subsystem 9-6

Applying RMA Guidelines to the Sonar Transmission Subsystem 9-11

Part 5 Appendices

Appendix A Rules of Thumb A-1

Appendix B Notation Used in This Handbook B-1

Appendix C Bibliography C-1

Appendix D Glossary D-1

Appendix E Index E-1



List of Techniques

Group 1 Getting a Sense for Schedulability 4-9

Technique 1 Using One Utilization Bound for the Entire Situation 4-10

Technique 2 Using Utilization Bounds for Each Event when Deadlines Are Within the Period 4-13

Group 2 Getting a Precise Schedulability Assessment 4-19

Technique 3 Sketching a Timeline 4-21

Technique 4 Calculating Response Time when Deadlines Are Within the Period 4-27

Technique 5 Calculating Response Time with Arbitrary Deadlines and Blocking 4-35

Technique 6 Calculating Response Time when Priorities Vary 4-42



Group 3 Assessing Capacity 4-54

Technique 7 Calculating Spare Capacity 4-55

Technique 8 Calculating Growth by Increasing Resource Usage of All Events 4-63

Technique 9 Eliminating Overrun 4-70

List of Situations

- Group 1** Handling Periodic Events 5-2
- Situation 1** Handling Periodic Events when Deadline Is Before or at the End of the Period 5-4
- Situation 2** Handling Periodic Events when Deadline Is After the End of the Period 5-15
- Group 2** Designing Tasks that Must Synchronize to Share Common Data 5-25
- Situation 3** Sharing a Single Data Resource 5-28
- Situation 4** Sharing Multiple Data Resources 5-44
- Group 3** Handling Aperiodic Events 5-61
- Situation 5** Handling Bounded Arrivals with Hard Deadlines 5-62
- Situation 6** Handling Bursty Arrivals with Hard Deadlines 5-93
- Situation 7** Handling Unbounded Arrivals with Soft Deadlines 5-115
- Group 4** Controlling Jitter 6-2
- Situation 8** Controlling Jitter on Input 6-5
- Situation 9** Controlling Jitter on Output 6-32
- Group 5** Message Passing Paradigms 6-64
- Situation 10** Using Sequential Message Handlers 6-66
- Situation 11** Using Parallel Message Handlers 6-82
- Group 6** Issues in Multiprocessor and Distributed Systems 6-94
- Situation 12** Determining End-to-End Resource Schedulability 6-100
- Situation 13** Handling Suspension of a Response 6-115
- Situation 14** Managing Priority Inversion in Distributed Systems 6-127
- Situation 15** Allocating and Prioritizing Tasks 6-133
- Group 7** New Events and Actions Introduced by Operating System Services 7-5
- Situation 16** New Events 7-8
- Situation 17** Actions at Priority of Calling Response 7-12
- Situation 18** Actions at a Higher Priority 7-14
- Situation 19** Atomic Actions on the CPU 7-19
- Situation 20** Implicit Operating System Resources 7-24



Group 8 Limited Representations 7-28

Situation 21 Coarse Time Granularity 7-29

Situation 22 Insufficient Number of Priorities 7-33

Group 9 Presence or Absence of Particular Operating System Features 7-38

Situation 23 Absolute and Relative Timers 7-40

Situation 24 Execution Capacities 7-45

Situation 25 Virtual Memory 7-47

Situation 26 Scheduling Policies 7-49

Situation 27 Synchronization Primitives 7-52

Situation 28 Message Passing Primitives 7-55


Group 10 Support for RMA Concepts in Standard Operating and Runtime Systems 7-60

Situation 29 Implementing Applications in POSIX-Compliant Operating Systems 7-62

Situation 30 Implementing Applications in Ada 7-85

List of Figures

- Figure 1-1:** Tables Used to Assess Schedulability 1-5
- Figure 2-1:** Periodic Tasks in Sample Problem 2-2
- Figure 2-2:** Unbounded Priority Inversion 2-5
- Figure 2-3:** Sample Problem with Blocking Added 2-6
- Figure 2-4:** Sample Problem with Aperiodics Added 2-9
- Figure 3-1:** Sequence Example 3-11
- Figure 4-1:** Timelines for Event $e1$ 4-23
- Figure 4-2:** Timeline for Events $e1$ and $e2$ 4-25
- Figure 4-3:** Response Time Technique, Step 1 4-31
- Figure 4-4:** Response Time Technique, Step 2 4-33
- Figure 4-5:** Response Time Technique, Step 3 4-34
- Figure 6-1:** Timeline with No Jitter from *Sleep_Until* 6-17
- Figure 6-2:** Timeline with Jitter from *Sleep_Until* 6-18
- Figure 6-3:** Timeline with Jitter and Phase Shift 6-18
- Figure 6-4:** Resources Used in a Distributed Response 6-96
- Figure 6-5:** Worst-Case Response Time for Action $a4$ of Event $e4$ when Jitter Is not Accounted for 6-105
- Figure 6-6:** Deferred Execution Effect Extends Response Time of $a4$ 6-106
- Figure 6-7:** Response Time of Event $e3$, Ignoring Suspension 6-118
- Figure 6-8:** Effect of Suspension on $e3$ Response Time 6-120
- Figure 7-1:** Schedulability Loss Versus the Number of Priority Bits 7-36
- Figure 7-2:** Use of intermediate Server Task to Implement Priority Inheritance 7-57

- 
- Figure 8-1:** Response Definition Legend 8-8
- Figure 8-2:** Response to *alpha 1* 8-8
- Figure 8-3:** Response to *alpha 2* 8-8
- Figure 8-4:** Response to *alpha 3* 8-9
- Figure 8-5:** Response to *charlie* 8-9
- Figure 8-6:** Response to *echo 1* and *echo 2* 8-9
- Figure 8-7:** Response to *fox 1* 8-10
- Figure 8-8:** Response to *fox 2* 8-10
- Figure 8-9:** Response to *gamma* 8-10
- Figure 8-10:** Response to *hector* 8-10
- Figure 8-11:** Response to *monkey* 8-10
- Figure 9-1:** Sonar Transmission Subsystem and Interfaces 9-6
- Figure 9-2:** Event Timing Illustration for the Sonar Transmit Sample Problem 9-7
- Figure 9-3:** High-Level Data and Control Flow Diagram 9-61

List of Tables

- Table 2-1:** Task Characteristics for Sample Problem 2-2
- Table 2-2:** Characteristics of Aperiodic Events for Sample Problem 2-9
- Table 2-3:** Summary of Sample Problem 2-13
- Table 3-1:** Summary of Event Sequence Types 3-6
- Table 3-2:** Summary of Event Sequence Arrivals 3-9
- Table 3-3:** Summary of Responses 3-15
- Table 3-4:** Summary of Resource Types and Policies 3-20
- Table 3-5:** Summary of Timing Requirements 3-22
- Table 3-6:** Symbols Used for Tables in this Section 3-23
- Table 3-7:** Syntax for Real-Time Situations 3-25
- Table 3-8:** Syntax for Arrivals 3-26
- Table 3-9:** Syntax for Responses 3-27
- Table 3-10:** Syntax for Resources 3-29
- Table 3-11:** Syntax for Timing Requirements 3-30
- Table 3-12:** Summary of Syntax 3-31
- Table 3-13:** Situation Table 3-35
- Table 3-14:** Areas Within a Situation Table 3-36
- Table 3-15:** Implementation Table 3-39
- Table 3-16:** Techniques Table 3-40
- Table 4-1:** Notation Used in This Handbook 4-1
- Table 4-2:** Functions Used in This Handbook 4-2
- Table 4-3:** Relationship Between Busy Period, Response Times, and Critical Instants 4-4
- Table 4-4:** Utilization Bounds for a Deadline Monotonic Priority Assignment 4-18
- Table 6-1:** Maximum Response Time for the Event Strings 6-72
- Table 6-2:** Maximum Response Time for the Event Handlers 6-77
- Table 6-3:** Maximum Response Time for the Event Handlers 6-81
- Table 6-4:** Maximum Response Time for the Event Strings 6-88
- Table 6-5:** Maximum Response Time for the Techniques Table Events 6-93

Table 7-1:	POSIX Standards—Base Standards	7-63
Table 7-2:	POSIX Standards—Additional Base Standards	7-63
Table 7-3:	POSIX Standards—Language Bindings	7-64
Table 7-4:	POSIX Standards—Application Environment Profiles	7-64
Table 7-5:	Characteristics of the Real-Time Profiles	7-66
Table 7-6:	RMA-Related Features and POSIX	7-74
Table 7-7:	RMA-Related Features and Ada83 and CIFO	7-91
Table 7-8:	RMA-Related Features and Ada9X	7-95
Table 8-1:	Six Steps for Analyzing a System	8-2
Table 8-2:	Execution Parameters for MP System Tasks	8-4
Table 8-3:	Arrival Rates of Environmental and Timed Events	8-6
Table 8-4:	Response Time Test Results for MP System	8-18
Table 8-5:	<i>H</i> Set Reduction Candidates	8-20
Table 8-6:	<i>HL</i> Set Reduction Candidates	8-21
Table 8-7:	<i>LH</i> Set Reduction Candidates	8-21
Table 8-8:	Reductions for <i>H</i> and <i>HL</i> Events	8-22
Table 8-9:	Results of New Application of Response Time Technique	8-24
Table 8-10:	Results of Response Time Test	8-32
Table 8-11:	Reduction Potentials Computed by Overrun Technique (<i>H</i> Set)	8-34
Table 8-12:	Reduction Potentials Computed by Overrun Technique (<i>LH</i> Set)	8-35
Table 8-13:	Reductions for <i>H</i> and <i>HL</i> Events	8-36
Table 8-14:	Response Time Test Results for Implementation 4	8-37
Table 8-15:	Response Time Test Results for Implementation 5	8-42
Table 9-1:	Completion Times for First Three Jobs of <i>T1</i> and <i>T2</i>	9-26
Table 9-2:	Completion Times for Jobs 1-10 of <i>Tc1</i>	9-33
Table 9-3:	Completion Times for <i>Tc1</i> Jobs	9-38
Table 9-4:	Completion Times of <i>Tc2</i> Jobs	9-39
Table 9-5:	Completion Times of <i>Tc2</i> Jobs When <i>Tc2</i> Priority Is Higher than <i>Tc1</i> Priority	9-41

Preface

About This Book

This handbook contains a collection of quantitative methods that enable real-time system developers to understand, analyze, and predict the timing behavior of many real-time systems. The methods are practical and theoretically sound, and can be used to assess design tradeoffs and to troubleshoot system timing behavior. We call this collection of methods **rate monotonic analysis (RMA)**.

This handbook has been created to serve as a definitive source of information and a guide for real-time developers as they analyze and design real-time systems.

RMA Originated with Rate Monotonic Scheduling Theory

A theoretical treatment of fixed-priority scheduling first appeared in 1973 when Liu and Layland [Liu 73] introduced the **rate monotonic scheduling algorithm** for independent periodic tasks. The term **rate monotonic** comes from a method of assigning priorities to a set of tasks in which priorities are assigned as a monotonic function of the rate of a periodic task: the higher the rate, the higher the priority. Given this simple rule for assigning priorities, Liu and Layland demonstrated properties for certain tasks when task deadlines coincide with the end of a task's period.

Theoretical Foundation of RMA

Rate monotonic scheduling theory has been extended from its original form of scheduling independent periodic tasks [Liu 73] to scheduling

- Both periodic and aperiodic tasks [Sprunt 89a].
- Tasks with synchronization requirements [Rajkumar 88], [Rajkumar 90], [Sha 90b].
- Tasks with mode change requirements [Sha 89a], [Tindell 92a].
- Tasks with deadlines before the end of the period [Leung 82].

Preface (Cont'd)

- Tasks with deadlines after the end of the period [Lehoczky 90].


In addition, the following issues have been addressed:

- Precise algorithms for determining schedulability [Joseph 86], [Lehoczky 89], [Audsley 90], [Lehoczky 90], [Audsley 91], [Harbour 91], [Tindell 92c], [Audsley 93].
- Associated hardware scheduling support [Lehoczky 86], [Sha 90c].
- Implications for Ada scheduling rules [Goodenough 88].
- Algorithm implementation in an Ada runtime system [Borger 89].
- Schedulability analysis of input/output paradigms [Klein 90].

Furthermore, design and analysis experiments have been performed to test the viability of the theory [Borger 89], [Locke 90]. Together, these findings constitute the set of analytical methods for real-time system engineering that we have named *rate monotonic analysis*.

**RMA
Assistance
Available**

We hope that you find RMA useful in designing, developing, and analyzing real-time systems, and we hope that this handbook facilitates your use of RMA. If you need information about a topic that is not covered or if there is a topic that you do not understand, we encourage you to contact Customer Relations at the Software Engineering Institute (SEI). The SEI will be happy to forward your question to an RMA consultant who will contact you as soon as possible.



Preface (Cont'd)

Suggestions for Improving the Handbook

In addition to providing a channel for assisting the readers of this book, we also would appreciate suggestions for improvement. If you find an error or omission, or have ideas for improving the clarity or usefulness of the handbook, please forward these to SEI Customer Relations.

Point of Contact

Customer Relations
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Phone: (412) 268-5800
Internet: customer-relations@sei.cmu.edu

Improvements

For a list of the improvements reflected in this printing, please contact SEI Customer Relations.

Acknowledgments

The authors gratefully acknowledge the contributions of the many people without whom this book would not have been possible.

The technology now known as rate monotonic analysis, upon which this book is based, had many key contributors. For a list of the important published findings in the field of rate monotonic analysis, please see the Bibliography.

We would like particularly to acknowledge the contributions of **Dr. Lui Sha** of the Software Engineering Institute and **Dr. John Lehoczky** of Carnegie Mellon University. With leadership, energy, and vision, Sha and Lehoczky transformed the fundamental findings of Liu and Layland [Liu 73] into a collection of methods and algorithms that we believe are of great value to real-time systems practitioners.

Dr. John B. Goodenough of the Software Engineering Institute has been the guiding force behind the transition of RMA technology into common practice. Goodenough's vision for RMA led directly to the Software Engineering Institute's RMA tutorials and to the production and ultimate publication of this handbook.

Jayne Cohen of Paramax contributed to the Glossary and provided an in-depth review of early drafts.

Reviewers

We further express our gratitude to the many people who were willing to take time from busy schedules to review and serve as test subjects for our early drafts: J. P. Rosen of ADALOG, France; Dr. Robert A. Brown of The Charles Stark Draper Laboratory, Inc.; Hassan Gomaa of George Mason University; Steve Vestal of Honeywell; Marion Moon of Hughes Aircraft; Dr. David Kirk, Thomas J. Mesler, and Dave Vogel of IBM; Pat Watson of IBM Federal Systems Co.; Tom McDermott of Lockheed Aeronautical Systems Co.; Steve Ignace, Don Kleinhans, Brian Newnam, Brian Pattison,

Acknowledgments (Cont'd)

and Steve Scheid of Magnavox; Milt Fulghum of McDonnell Douglas ESC; David B. Rodman of MITRE; Brad Clark, Mary Jo Haselhorst, Del Killian, Keith Kohout, Lee Lucas, Arlo Mickelsen, Bob Page, Tom Roseman, Bob Westbrook, and John Zenor of the Naval Air Warfare Center, China Lake; Bob Sedlmeyer and Dr. David J. Thuente of Indiana University-Purdue University, Ft. Wayne; Jim Murphy of Raytheon; John Hammer of Search Technology; Pat Rogers of Software Arts & Sciences; Patricia Brannigan, Larry Howard, Dr. Craig Meyers, Patrick Donohoe, Dr. Ragunathan Rajkumar, Michael Rissman, Rob Veltre, and Dr. Nelson Weideman of the Software Engineering Institute; Mike Cochran and Richard A. Kirk of Software Productivity Consortium; Dwight Fohner of Sverdrup; Mr. Gentry Gardner of Telos Corporation; Stefan Fuchs of Universität Karlsruhe, Germany; Art Moorshead of Westinghouse; and John Barnes (consultant).

We also wish to thank those who attended the review session for our Beta edition: Godfrey Vassallo of Paramax; Russ Plain and Peter Kortmann of Tri-Pacific; and Dr. Gary Chastek of the Software Engineering Institute.

We would like to give a special thanks to those reviewers who tirelessly endured a review of the final version of the handbook: Dr. John B Goodenough, Lee Lucas, Dr. Ragunathan Rajkumar, Dr. David J. Thuente, and Dr. Nelson Weideman. Thanks also to Dr. Daniel Katcher for help in correcting errors in the first printing.

Production and Editorial Assistants

At the Software Engineering Institute, Barbara Gratz and Barbara Zayas provided many hours of production and editorial assistance. Additional support was provided by Susan Abdalla, Daniel Bidwa, Todd Bowman, Pamela Burger-Hughes, Robert Lang, Mary McElhinney, Roberta Schwerer, Skip Shelly, and Mary Zoys.