
Undergraduate Topics in Computer Science

Series editor

Ian Mackie

Advisory board

Samson Abramsky, University of Oxford, Oxford, UK

Karin Breitman, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil

Chris Hankin, Imperial College London, London, UK

Dexter Kozen, Cornell University, Ithaca, USA

Andrew Pitts, University of Cambridge, Cambridge, UK

Hanne Riis Nielson, Technical University of Denmark, Kongens Lyngby, Denmark

Steven Skiena, Stony Brook University, Stony Brook, USA

Iain Stewart, University of Durham, Durham, UK

Undergraduate Topics in Computer Science (UTiCS) delivers high-quality instructional content for undergraduates studying in all areas of computing and information science. From core foundational and theoretical material to final-year topics and applications, UTiCS books take a fresh, concise, and modern approach and are ideal for self-study or for a one- or two-semester course. The texts are all authored by established experts in their fields, reviewed by an international advisory board, and contain numerous examples and problems. Many include fully worked solutions.

For further volumes:

<http://www.springer.com/series/7592>

Philippe Lalanda • Julie A. McCann
Ada Diaconescu

Autonomic Computing

Principles, Design and Implementation

 Springer

Philippe Lalanda
Laboratoire Informatique de Grenoble
Université Joseph Fourier
Grenoble, France

Julie A. McCann
Department of Computing
Imperial College London
London, UK

Ada Diaconescu
Department of Computing and Networking
Télécom ParisTech
Paris, France

ISSN 1863-7310
ISBN 978-1-4471-5006-0 ISBN 978-1-4471-5007-7 (eBook)
DOI 10.1007/978-1-4471-5007-7
Springer London Heidelberg New York Dordrecht

Library of Congress Control Number: 2013936543

© Springer-Verlag London 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

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

Foreword

The autonomic computing initiative—the creation of self-managing systems—was a call to arms from IBM (Paul Horn, 2001) to the industry and academic computing communities and signified the mainstream adoption of such research fields as intelligent fault management.

Although IBM set it in the context of coping with the ever-increasing ‘systems of systems’ complexity and dealing with the total cost of ownership (TCO), researchers expanded this to next generation of computing and communications, for instance, pervasive computing, ubiquitous computing, swarm-based computing, agent-based computing, smart grid, on-demand computing, next-generation Internet, adaptive communications and the latest trends, green and cloud computing. These streams of research have found focus through such research conferences as ICAC, EASe, SEAMS, MUCS and SASO (International Conference on Autonomic Computing, Engineering of Autonomic and Autonomous Systems, Software Engineering for Adaptive and Self-Managing Systems, Managing Ubiquitous Communication and Services and Self-Adaptive and Self-Organizing Systems, respectively). The decade of research also saw the creation of related initiatives such as Autonomic Communications and Organic Computing, as well as communities, such as the IEEE Technical Committee on Autonomous and Autonomic Systems (TCAAS) and the Autonomic Communications Forum (ACF).

Due to the complexity of the challenges that this subject brings to the fore, the Autonomic Computing has to rely on expertise from many fields—software engineering, systems engineering, control theory and AI (artificial intelligence), among others. As such it is hard to ‘nail’ down the field which is possibly the reason there hasn’t emerged a good tutorial text—till now. As with many ‘hot’ research initiatives, the spotlight (and often the funding) moves on, yet the original long-term 2020–2030 needs have not. If anything, they have become more critical. The Software Crisis 2.0 (Fitzgerald 2012) highlighted more than ever the need to have our software self-managing, due to the demand for data from digital natives, coupled with the huge volume of data now generated through ubiquitous mobile devices, sensors and applications.

The success of the initiative has already been indicated by the notable move of ‘autonomicity’ from the previously mentioned conferences and communities to a standard topic in almost all computer- and communications-based conferences and communities. Ironically, the final success of the initiative may be marked as it no longer exists separately as a specialised field but as a standard, invisible and integrated part of our systems and software engineering.

For the autonomic systems research and development to make further leaps and bounds and move convincingly into the next decade to meet the Software Crisis 2.0 and its other longer term goals requires that it move beyond the research labs and PhD programmes to our graduate, undergraduate and CPPD (continuous professional and personal development) courses.

This book marks the enabler for that next stage.

University of Ulster, Northern Ireland
12 December 2012

Roy Sterritt

References

- Fitzgerald, B.: Software crisis 2.0. *IEEE Comput.* **45**(4) (2012)
Horn, P.: *Autonomic computing: IBM’s perspective on the state of information technology*. IBM T.J. Watson Labs, New York (2001)

Preface

Autonomic computing seeks to render computing systems as self-managed. In other words, its objective is to enable computer systems to manage themselves so as to minimise the need for human input. Autonomic computing as an approach is guaranteed to change the way software systems are developed. Indeed, this new field is addressing some of the issues resulting from the ever-increasing complexity of software administration and the growing difficulty encountered by software administrators in performing their job effectively.

This book provides a practical perspective on autonomic computing. Implementing self-managed systems remains a true challenge today. Thus, beyond giving necessary explanations about the objectives and interests of autonomic computing, this book goes through the different software engineering techniques that are currently available for organising and developing self-managed software systems. In summary, this book uniquely:

- Provides a structured and comprehensive introduction to autonomic computing with a software engineering perspective, as far as we are aware this is the first book to do so
- Presents highly up-to-date information on techniques implementing self-monitoring, self-knowledge, decision-making and self-adaptation
- Provides a downloadable learning environment and source code that allows students to develop, execute and test autonomic applications at an associated website

Authors have created the aforementioned learning environment and placed it on a web page that will be regularly updated. The environment represents an autonomic pervasive computing application that simulates a *digital home*. A dedicated development environment has been designed around this; it allows the student to execute autonomic code in a runtime simulation that provides concrete, visual feedback of the behaviours illustrating what the student has programmed.

This book is aimed at students and practitioners working on software projects where system self-management would redress maintenance complexity and cost issues. Several aspects of this book have been tested in a classroom, which makes this book ideal for a 10-week lecture programme.

Content Level: master student/professional

Keywords: Autonomic computing, Software engineering, Software architectures, Software monitoring, Software adaptation, Knowledge and reasoning

Related Subjects: Software engineering

Authors: Authors are practitioners and recognised researchers in the field. They have published more than 200 publications in international conferences and journals.

Grenoble, France
London, UK
Paris, France

Philippe Lalanda
Julie A. McCann
Ada Diaconescu

Acknowledgments

Many useful discussions with colleagues and students helped in the preparation of this book. The authors would like to thank the following people who, without reward, reviewed and critiqued the text; their comments and suggestions have been invaluable in ensuring the quality of this book:

Luciano Baresi, Charles Consel, Clément Escoffier, Catherine Hamon, Roman Kolcun, Pedro Martins, Iulian Neamtiu, Simon O’Keefe, Alessandra Russo, Poonam Yadav and Shusen Yang.

We are also grateful to Simon Rees of Springer who encouraged us to write the book and provided invaluable assistance in the production of the final copy.

We would also like to thank our colleagues, friends and family for their constant support, encouragement and patience. Ada thanks Mr. Smith for regularly changing the subject. Julie thanks husband Grant and son Carter—you can now use my laptop to watch 1950s cartoons. Philippe thanks his wife, now an expert in autonomic computing, and his two sons, Grégoire and Arthur—experts to come!

Contents

1	Software Engineering to Autonomic Computing	1
1.1	Software Complexity	1
1.2	The Software Life Cycle	5
1.2.1	Software Development.....	5
1.2.2	Software Deployment	7
1.2.3	Software Maintenance	9
1.3	Maintenance Challenges	11
1.4	Autonomic Computing	14
1.5	Book Structure	17
1.6	Key Points	19
	References.....	20
2	Autonomic Systems	23
2.1	Autonomic Computing	24
2.1.1	Definitions.....	24
2.1.2	Goals	26
2.1.3	Context.....	28
2.2	The Origins and Motivations Behind Autonomic Computing	30
2.3	Self-* Properties and Expected Qualities	34
2.3.1	Autonomic Key Features	34
2.3.2	Fundamental Self-* Features	35
2.3.3	Extended Self-* Capabilities	37
2.4	Benefits, Challenges and Degrees of Autonomy	39
2.4.1	Benefits of Autonomic Computing	39
2.4.2	Challenges of Autonomic Computing	40
2.4.3	An Incremental Approach to Autonomic Computing	41
2.5	Similar Initiatives, Current Status and Relation to Software Engineering	43
2.5.1	Autonomicity in Existing Systems	43
2.5.2	Top-Down Versus Bottom-Up Approaches	43
2.5.3	Similar Initiatives.....	44

2.6	Key Points.....	52
	References.....	54
3	Sources of Inspiration for Autonomic Computing.....	57
3.1	Overview of Influences.....	58
3.1.1	Introduction.....	58
3.1.2	Natural Systems.....	59
3.1.3	Adaptive Computing Systems.....	61
3.2	Biology.....	63
3.2.1	Overview.....	63
3.2.2	Introduction to Biological Nervous Systems.....	64
3.2.3	Structure of the Human Nervous System.....	65
3.2.4	Function of the Human Nervous System.....	67
3.2.5	Reflexes and Autonomic Control Loops.....	70
3.2.6	Different Nervous System Architectures and Features.....	71
3.2.7	Summary of Inspiration from Nervous Systems.....	72
3.2.8	Bio-inspiration Beyond Nervous Systems.....	73
3.3	Control Systems.....	74
3.3.1	Introduction.....	75
3.3.2	Feedback Control.....	77
3.3.3	The PID Controller.....	78
3.3.4	Oscillations, Overshooting, Damping and Stability.....	80
3.3.5	Control and Autonomic Computing.....	81
3.4	Artificial Intelligence.....	82
3.4.1	Introduction to Intelligence.....	82
3.4.2	Introduction to Software Agents.....	84
3.4.3	Building Artificial Intelligence.....	86
3.4.4	Summary of AI Relevance for AC.....	89
3.5	Complex Systems.....	89
3.6	Key Points.....	91
	References.....	92
4	Autonomic Computing Architectures.....	95
4.1	Autonomic Elements.....	96
4.2	Architecture of Autonomic Elements.....	98
4.2.1	IBM Reference Architecture.....	98
4.2.2	Sensors.....	100
4.2.3	Effectors.....	102
4.2.4	Autonomic Manager.....	104
4.2.5	Architectural Properties of Autonomic Managers.....	106
4.3	Autonomic Manager Reference Architecture.....	109
4.3.1	The MAPE-K Model.....	109
4.3.2	Monitoring.....	111
4.3.3	Analysis.....	113
4.3.4	Planning.....	115

4.3.5	Execution	117
4.3.6	Summary	120
4.4	Architecture with Multiple Autonomic Elements.....	121
4.4.1	Introduction.....	121
4.4.2	Hierarchical <i>Versus</i> Decentralised Organisation.....	122
4.4.3	The ANS Example	124
4.5	Key Points.....	126
	References.....	127
5	The Monitoring Function	129
5.1	Introduction to Monitoring	130
5.2	Performance Monitoring.....	132
5.3	Knowing What to Monitor and Monitoring Overheads.....	134
5.4	Profiling	137
5.5	Monitoring Overheads	138
5.6	Monitoring for <i>Free</i>	139
5.7	Building Probes	141
5.8	Examples of Monitoring Tools, Frameworks and Platforms	145
5.9	Monitoring the Monitors: Adaptive Monitoring.....	148
5.10	Key Points.....	149
	References.....	150
6	The Adaptation Function	153
6.1	Software Adaptation	154
6.2	Code Adaptation	156
6.2.1	Upgrading Code.....	156
6.2.2	Integrating Code	159
6.3	Code Adaptation Techniques	161
6.3.1	OS-Level Adaptation	161
6.3.2	Program-Level Adaptation.....	163
6.3.3	Component-Level Adaptation.....	168
6.3.4	Software Services	170
6.4	OSGi	173
6.4.1	Modularity	173
6.4.2	Service	175
6.4.3	Conclusion	177
6.5	iPOJO.....	177
6.6	Conclusion	181
6.7	Key Points.....	182
	References.....	182
7	The Decision Function	185
7.1	Introduction to Knowledge	186
7.1.1	Definition	186
7.1.2	Forms of Knowledge.....	187
7.1.3	Knowledge Representation	189

7.2	Knowledge in Autonomic Managers	190
7.2.1	Introduction.....	190
7.2.2	Rule-Based Autonomic Systems	191
7.2.3	Model-Based Autonomic Systems	193
7.2.4	Goal-Based Autonomic Systems	194
7.2.5	Utility-Based Autonomic Systems	196
7.2.6	Autonomic Systems That Learn	197
7.3	Model-Driven Autonomicity	199
7.3.1	Introduction.....	199
7.3.2	Model Representation.....	200
7.3.3	Architectural Models	201
7.4	Reasoning Techniques	204
7.4.1	Programming Languages	204
7.4.2	Search-Based Reasoning	205
7.4.3	Logic-Based Reasoning.....	206
7.4.4	Classifiers and Statistical Learning Methods.....	208
7.5	Bayesian Networks Example.....	208
7.6	Key Points.....	213
	References.....	214
8	Evaluation Issues.....	217
8.1	Evaluating Autonomic Systems.....	218
8.2	Evaluation Elements	219
8.2.1	Quality of Service	219
8.2.2	Cost	219
8.2.3	Adaptivity	221
8.2.4	Time to Adapt and Reaction Time	221
8.2.5	Sensitivity	222
8.2.6	Stabilisation	223
8.2.7	Failure Avoidance (Robustness) and Autonomy	224
8.2.8	Interfacing to the Outside World	225
8.2.9	Centralisation Versus Decentralisation	226
8.2.10	Granularity/Flexibility	227
8.3	Some Evaluation Metrics for Emergent Systems	228
8.3.1	Price of Anarchy (PoA)	229
8.3.2	Equilibrium.....	229
8.4	Benchmarking.....	230
8.5	The Autonomic Computing Benchmark: A Summary	232
8.6	Key Points.....	233
	References.....	234
9	Autonomic Mediation in Cilia.....	235
9.1	Software Integration	236
9.2	Cilia.....	239
9.3	Autonomic Cilia.....	242
9.3.1	Overview.....	242
9.3.2	Cilia Touchpoints	243

9.3.3	Cilia Meta-level and Base Level.....	245
9.3.4	Cilia Dynamic Monitoring.....	247
9.3.5	Cilia Dynamic Adaptation.....	249
9.3.6	Knowledge Module.....	251
9.4	Towards Autonomic Life-Cycle Management of Cilia Chains.....	251
9.4.1	Challenges and Motivation.....	251
9.4.2	Model-Based Solutions.....	253
9.4.3	The Cube Project.....	254
9.5	Key Points.....	259
	References.....	260
10	Future of Autonomic Computing and Conclusions	263
10.1	Autonomic Computing in This Book.....	264
10.2	Alternative Autonomic Stories.....	265
10.2.1	Autonomic Communications.....	265
10.2.2	Autonomic Computing, Right Down to the Metal?.....	268
10.3	Autonomic Computing in the Near Future.....	270
10.3.1	Engineering Autonomic Systems.....	271
10.3.2	Managing Complexity.....	272
10.3.3	Who Guards the Guards? Trust and Assurances in Autonomic Computing.....	273
10.4	Conclusion.....	276
10.5	Key Points.....	277
	References.....	278
	Annex: Learning Environment	279
	Index	285