

SpringerBriefs in Computer Science

Series Editors

Stan Zdonik
Peng Ning
Shashi Shekhar
Jonathan Katz
Xindong Wu
Lakhmi C. Jain
David Padua
Xuemin Shen
Borko Furht
V. S. Subrahmanian
Martial Hebert
Katsushi Ikeuchi
Bruno Siciliano

For further volumes:
<http://www.springer.com/series/10028>

Katia Potiron · Amal El Fallah Seghrouchni
Patrick Taillibert

From Fault Classification to Fault Tolerance for Multi-Agent Systems

Katia Potiron
Thales Airborne Systems
Louvenciennes, Yvelines
France

Patrick Taillibert
Thales Airborne Systems
Louvenciennes, Yvelines
France

Amal El Fallah Seghrouchni
LIP6, University Pierre and Marie Curie
Paris
France

ISSN 2191-5768
ISBN 978-1-4471-5045-9
DOI 10.1007/978-1-4471-5046-6
Springer London Heidelberg New York Dordrecht

ISSN 2191-5776 (electronic)
ISBN 978-1-4471-5046-6 (eBook)

Library of Congress Control Number: 2013932457

© The Author(s) 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)

Preface

This book presents a summary of the research achieved by Katia Potiron during her Ph.D. Thesis defended on 14 April 2010 at the University Paris 6. This thesis was co-supervised by Patrick Taillibert, expert engineer at Thales Airborne Systems and Amal El Fallah Seghrouchni, Professor at the University Paris 6 (University Pierre and Marie Curie, France).

The intuition of this research is that, thanks to the Multi-Agent Systems (MAS) approach, fault tolerance property might be achieved naturally for complex software and should be complementary with classical existing approaches. This book, as a one-year “night work” by Katia Potiron after her thesis defense, tries to summarize the thesis contributions and to provide a comprehensive view of this research. The book contains additional explanations to make accessible to those not familiar with the subject the ideas sustaining the thesis propositions.

From a research point of view, this work is a cross-disciplinary attempt between the well-established field of fault tolerance and the emerging field of MAS issued from distributed artificial intelligence. Indeed, the MAS paradigm plays today an important role in complex software development. The associated technology offers a large panel of original concepts, architectures, interaction protocols, and methodologies for the analysis and the specification of complex systems built as MAS. One of the driving motivations for this work is the observation that MAS, as a technology, still lacks mechanisms to guarantee robustness and fault tolerance properties. These properties are crucial from a software perspective, especially when MAS are built for critical or military applications where dependability is vital. The expected properties vary according to the effects of the abnormal behavior of the software on the system safety, what is represented, for example, by design assurance level for software in civil airborne systems but this aspect is not studied here since it is related to a third domain that would be safety assessment. Hence, this book tries to emphasize the characterization of MAS with regard to existing studies in fault tolerance domain.

For classical systems, a fault classification exists and allows defining faults. So that, when dependability is at stake, such a fault classification may be used, from the beginning of the system design, to define fault classes and specify which types of faults are expected for the system and the software. Thus, one may tend to use such a fault classification for MAS, but the fact that agents are autonomous and

proactive may come into consideration on the faults potentially occurring in the system. As a matter of fact, this kind of behavior is not taken into account in the present fault classification. Moreover, autonomous and proactive agents are primarily “intelligent agents”. Does this “intelligence” have a role to play with regard to fault tolerance? Is it possible to take advantage of an agent property to obtain a more effective fault handler? Or are the agent properties an impediment to fault tolerance?

In addition, the field of fault tolerance provides numerous methods adapted to handle different kinds of faults. Some handling methods had been studied in the Multi-Agent System domain, adapting to their specificities and capabilities but at the same time increasing the large amount of fault tolerance methods to consider. Therefore, unless one is an expert in fault tolerance, it is difficult to choose, evaluate, or compare fault tolerance methods. This prevents many applications from using these methods and, consequently, to be tolerant to common faults. That is the reason why this book also tries to answer the important question of how to derive some guidelines and fault handlers based on the fault classification and the MAS studies (for instance from the properties specification phase).

Finally, the authors would like to thank Gilles Klein for his insightful review of this book, Costin Caval for his review and future work on handling unforeseen faults, and Nicolas Viollette for his long-term support to Katia Potiron. We also address a special acknowledgment to Professor Karin K. Breitman who made possible the publication of this thesis.

December 2012

Katia Potiron
Amal El Fallah Seghrouchni
Patrick Taillibert

Contents

1	Introduction	1
	References	3
2	Multi-Agent System Properties	5
2.1	Multi-Agent System and Agents	5
2.2	Agents and Faults	7
2.3	Agents and Fault Tolerance	8
	References	10
3	Fault Classification	11
3.1	Conventional Fault Classification	11
3.2	Existing Faults in MAS	15
3.2.1	Conventional Faults Relevance for MAS	15
3.2.2	Reported Handled Faults in MAS	15
	References	18
4	Refinement of the Fault Classification for MAS	21
4.1	The First Attribute Issue	21
4.2	Agent Centered Analysis	23
4.3	System Centered Analysis	26
4.4	Faults Review	28
4.5	Validity of This Approach	29
4.5.1	Faults Comparison	29
4.5.2	Analysis of the Difference Between Faults	33
4.6	Preliminary Conclusion	34
	References	35
5	Fault Tolerance for MAS Specific Faults	37
5.1	Context	37
5.1.1	Faults Leading to EMP	38
5.1.2	Consequences	38
5.2	Related Work	39

5.3	A Handler for the Agents	41
5.3.1	Analysis of the EMP	41
5.3.2	Solution Key Idea	42
5.3.3	Formal Description	42
5.3.4	Conversation Formalism	43
5.4	Effectiveness.	47
5.5	Using the Resend Performative	49
5.5.1	Integration Resend into Agent Conversations	49
5.5.2	Integration in the Agents	54
5.6	Discussion on Some Aspects of Resend	54
5.6.1	Resend Only Once	55
5.6.2	The Case of Event Perception Instead of a Message	55
5.6.3	Experiments Feedback	55
	References	56
6	Fault Classification Attributes as an Ontology to Build Fault Tolerant MAS.	59
6.1	Fault Specification Phase	59
6.1.1	The Specification Phase of the Entire System.	60
6.1.2	The Dispatching Phase Among the System Entities.	62
6.1.3	Comments on the Specifications	65
6.2	Handler Specification Phase	66
6.2.1	Handlers Comparison: An Example.	66
6.2.2	Analysis of Handlers and Detection Means	70
6.2.3	Choice and Monitoring of Handlers	71
	References	73
7	Conclusion.	75
7.1	Summary	75
7.2	Future Work	76
7.2.1	Unforeseen Faults	76
7.2.2	Diagnosis for the Choice of a Fault Handler	77
7.2.3	Resend Automation	77
7.2.4	Prospects About Persistent Faults	78
	References	78