Artificial Intelligence Methods *in* Software Testing

SERIES IN MACHINE PERCEPTION AND ARTIFICIAL INTELLIGENCE*

Editors: **H. Bunke** (Univ. Bern, Switzerland)
P. S. P. Wang (Northeastern Univ., USA)

- Vol. 43: Agent Engineering
(Eds. *Jiming Liu, Ning Zhong, Yuan Y. Tang and Patrick S. P. Wang*)
- Vol. 44: Multispectral Image Processing and Pattern Recognition
(Eds. *J. Shen, P. S. P. Wang and T. Zhang*)
- Vol. 45: Hidden Markov Models: Applications in Computer Vision
(Eds. *H. Bunke and T. Caelli*)
- Vol. 46: Syntactic Pattern Recognition for Seismic Oil Exploration
(*K. Y. Huang*)
- Vol. 47: Hybrid Methods in Pattern Recognition
(Eds. *H. Bunke and A. Kandel*)
- Vol. 48: Multimodal Interface for Human-Machine Communications
(Eds. *P. C. Yuen, Y. Y. Tang and P. S. P. Wang*)
- Vol. 49: Neural Networks and Systolic Array Design
(Eds. *D. Zhang and S. K. Pal*)
- Vol. 50: Empirical Evaluation Methods in Computer Vision
(Eds. *H. I. Christensen and P. J. Phillips*)
- Vol. 51: Automatic Diatom Identification
(Eds. *H. du Buf and M. M. Bayer*)
- Vol. 52: Advances in Image Processing and Understanding
A Festschrift for Thomas S. Huwang
(Eds. *A. C. Bovik, C. W. Chen and D. Goldgof*)
- Vol. 53: Soft Computing Approach to Pattern Recognition and Image Processing
(Eds. *A. Ghosh and S. K. Pal*)
- Vol. 54: Fundamentals of Robotics — Linking Perception to Action
(*M. Xie*)
- Vol. 55: Web Document Analysis: Challenges and Opportunities
(Eds. *A. Antonacopoulos and J. Hu*)
- Vol. 56: Artificial Intelligence Methods in Software Testing
(Eds. *M. Last, A. Kandel and H. Bunke*)
- Vol. 57: Data Mining in Time Series Databases
(Eds. *M. Last, A. Kandel and H. Bunke*)
- Vol. 58: Computational Web Intelligence: Intelligent Technology for Web Applications
(Eds. *Y. Zhang, A. Kandel, T. Y. Lin and Y. Yao*)
- Vol. 59: Fuzzy Neural Network Theory and Applications
(*P. Liu and H. Li*)

*For the complete list of titles in this series, please write to the Publisher.

Artificial Intelligence Methods *in* Software Testing

Editors

Mark Last

Ben-Gurion University of the Negev, Israel

Abraham Kandel

Tel-Aviv University, Israel

University of South Florida, Tampa, USA

Horst Bunke

University of Bern, Switzerland

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI

Published by

World Scientific Publishing Co. Pte. Ltd.

5 Toh Tuck Link, Singapore 596224

USA office: Suite 202, 1060 Main Street, River Edge, NJ 07661

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

ARTIFICIAL INTELLIGENCE METHODS IN SOFTWARE TESTING

Series in Machine Perception and Artificial Intelligence (Vol. 56)

Copyright © 2004 by World Scientific Publishing Co. Pte. Ltd.

All rights reserved. This book, or parts thereof, may not be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system now known or to be invented, without written permission from the Publisher.

For photocopying of material in this volume, please pay a copying fee through the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, USA. In this case permission to photocopy is not required from the publisher.

ISBN 981-238-854-0

Printed in Singapore by World Scientific Printers (S) Pte Ltd

Dedicated to

The Honorable Congressman C. W. Bill Young

House of Representatives

For his vision and continuous support in creating the National Institute
for Systems Test and Productivity at the Computer Science and
Engineering Department, University of South Florida

This page is intentionally left blank

Preface

Over the last 50 years, Artificial Intelligence (AI) researchers have developed a variety of computational models and methods that can make a computer program to behave in a reasonably “intelligent” way, i.e. to imitate such routine human tasks as learning, vision, speech recognition, etc. Numerous implementations of AI-based methods range from Optical Character Recognition (OCR) software to autonomous robots. Despite an apparent discrepancy between the way our brains think and the way computers work, the combination of computer power with human-like reasoning has generated impressive results in diverse domains. The purpose of this book is to shed the light on a relatively new and promising application area of AI techniques, namely *software testing*.

Software testing at various levels (unit, component, integration, etc.) has become a major activity in systems development. Though a significant amount of research has been concerned with formal proofs of code correctness, development teams still have to validate their increasingly complex products by executing a large number of experiments, which imitate the actual operating environment of the tested system. And, as we all know, the result is hardly satisfactory: commercial software products are abundant with undocumented “features”, which are neither required nor expected by any rational customer. A failure to meet the stated requirements is usually caused by a design flaw, or a “bug”. Discovering the hidden faults rather than proving that those do not exist in software is, in fact, the main purpose of software tests.

The state-of-the-art in software testing has a grave impact on the world’s economy. Thus, leading software companies are forced to employ one tester per each active coder. Another industry trend is to transfer the testing activities to countries with cheap high-tech labor. A 2002 study by the US National Institute of Standards & Technology has found that “the national annual costs of an inadequate infrastructure for software testing is estimated to range from \$22.2 to \$59.5 billion” or about 0.6 percent of the US gross domestic product. This number does not include costs associated with catastrophic failures of mission-critical software such as the \$165 million Mars Polar Lander shutdown in 1999 which was attributed to a software glitch. According to another report, U.S. Department of Defense alone loses over four billion dollars a year due to software failures.

The characteristics of software testing are quite similar to other problems successfully tackled by AI techniques. Thus, we need to find an optimal set of

actions (tests) subject to resource constraints (limited testing resources such as time) under uncertainty with respect to the number, severity, and location of undetected bugs. It is also important to understand the inherent probabilistic nature of software testing: we cannot require a test object to be “bug-free”; at most, we can assume that its likelihood to fail while in use is sufficiently low considering the criticality of the corresponding failure.

The authors of this book’s chapters demonstrate the application of AI-based methods to various aspects of the software testing process as follows.

Chapter 1 by Witold Pedrycz and George Vukovich presents the fuzzy set-based approach to cause-effect software modeling as a basis for designing a test oracle in black-box software testing. Such an oracle can be used to predict the outcome of a given test.

A tested object can be executed an unlimited number of times thus generating large amounts of execution data. **Chapter 2** by Mark Last and Menahem Friedman demonstrate the potential use of data mining algorithms for automated induction of functional requirements from such data. In addition to being a test oracle, the resulting model can be utilized for designing a minimal set of test cases.

Data-driven techniques for automated testing are not applicable to Graphical User Interface (GUI) software, which is characterized by event-driven input and graphical output. A novel method for automated GUI regression testing using AI planning is presented by Atif M. Memon in **Chapter 3**. The proposed technique is based on re-planning GUI test cases by associating a task with each test case.

Artificial Neural Networks (ANN) is a well established method for learning complex functions from training examples. **Chapter 4** by Prachi Saraph, Mark Last, and Abraham Kandel shows the use of neural networks for automated input-output analysis of data-driven programs. The proposed methodology is based on advanced algorithms for efficient network pruning.

Effectiveness of software testing can be increased dramatically if one can estimate the quality of each system module prior to test design and execution. In **Chapter 5**, Taghi M. Khoshgoftaar and Naeem Seliya present the three-group classification model for predicting the number of faults as a function of common software metrics.

The distribution of faults across system modules may be *skewed* meaning that only a few complex modules are prone to failure. **Chapter 6** by Scott Dick and Abraham Kandel uses the machine learning technique of resampling to overcome the problem of skewness in software metrics databases. The resampling methodology can improve the performance of any data mining algorithm applied to software testing data.

We believe that the chapters collected in this book present only a small, but representative sample of potential AI applications in the extremely important

areas of software testing and software quality assurance. As more researchers will contribute to this challenging field, we expect to see evolvement of practical AI-based tools for automated testing of complex software systems.

Acknowledgments

We thank all the contributing authors who, despite their busy schedule, have responded enthusiastically to our invitation and have done such an excellent job on their chapters. We would like to thank Ronit Levy who has done final formatting of all chapters. Our special thanks go to Editor Ian Selstrup for his prompt and patient attitude. The preparation of this volume was partially supported by the National Institute for Systems Test and Productivity at the University of South Florida under U.S. Space and Naval Warfare Systems Command grant number N00039-01-1-2248. We would also like to acknowledge the partial support provided by the Fulbright Foundation to Prof. Kandel. During the academic year 2003-2004 Prof. Kandel has been awarded Fulbright Research Award at Tel-Aviv University, College of Engineering.

September 2003

*Mark Last
Abraham Kandel
Horst Bunke*

This page is intentionally left blank

Contents

Preface.....	vii
Chapter 1 Fuzzy Cause – Effect Models of Software Testing	1
<i>Witold Pedrycz and George Vukovich</i>	
Chapter 2 Black-Box Testing with Info-Fuzzy Networks	21
<i>Mark Last and Menahem Friedman</i>	
Chapter 3 Automated GUI Regression Testing Using AI Planning	51
<i>Atif M. Memon</i>	
Chapter 4 Test Set Generation And Reduction With Artificial Neural Networks.....	101
<i>Prachi Saraph, Abraham Kandel, and Mark Last</i>	
Chapter 5 Three-Group Software Quality Classification Modeling Using An Automated Reasoning Approach	133
<i>Taghi M. Khoshgoftaar and Naeem Seliya</i>	
Chapter 6 Data Mining with Resampling in Software Metrics Databases	175
<i>Scott Dick and Abraham Kandel</i>	