COMPUTATIONAL INTELLIGENCE IN SOFTWARE ENGINEERING

ADVANCES IN FUZZY SYSTEMS — APPLICATIONS AND THEORY

Honorary Editor: Lotfi A. Zadeh (*Univ. of California, Berkeley*)

Series Editors: Kaoru Hirota (*Tokyo Inst. of Tech.*),
George J. Klir (*Binghamton Univ. — SUNY*),
Elie Sanchez (*Neurinfo*),
Pei-Zhuang Wang (*West Texas A&M Univ.*),
Ronald R. Yager (*Iona College*)

-
- Vol. 1: Between Mind and Computer: Fuzzy Science and Engineering
(Eds. P.-Z. Wang and K.-F. Loe)
- Vol. 2: Industrial Applications of Fuzzy Technology in the World
(Eds. K. Hirota and M. Sugeno)
- Vol. 3: Comparative Approaches to Medical Reasoning
(Eds. M. E. Cohen and D. L. Hudson)
- Vol. 4: Fuzzy Logic and Soft Computing
(Eds. B. Bouchon-Meunier, R. R. Yager and L. A. Zadeh)
- Vol. 5: Fuzzy Sets, Fuzzy Logic, Applications
(G. Bojadziev and M. Bojadziev)
- Vol. 6: Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers
by Lotfi A. Zadeh
(Eds. G. J. Klir and B. Yuan)
- Vol. 7: Genetic Algorithms and Fuzzy Logic Systems: Soft Computing
Perspectives
(Eds. E. Sanchez, T. Shibata and L. A. Zadeh)
- Vol. 8: Foundations and Applications of Possibility Theory
(Eds. G. de Cooman, D. Ruan and E. E. Kerre)
- Vol. 9: Fuzzy Topology
(Y. M. Liu and M. K. Luo)
- Vol. 10: Fuzzy Algorithms: With Applications to Image Processing and
Pattern Recognition
(Z. Chi, H. Yan and T. D. Pham)
- Vol. 11: Hybrid Intelligent Engineering Systems
(Eds. L. C. Jain and R. K. Jain)
- Vol. 12: Fuzzy Logic for Business, Finance, and Management
(G. Bojadziev and M. Bojadziev)
- Vol. 13: Fuzzy and Uncertain Object-Oriented Databases: Concepts and Models
(Ed. R. de Caluwe)
- Vol. 15: Fuzzy-Logic-Based Programming
(Chin-Liang Chang)

Forthcoming volumes:

- Vol. 14: Automatic Generation of Neural Network Architecture Using
Evolutionary Computing
(Eds. E. Vonk, L. C. Jain and R. P. Johnson)

Advances in Fuzzy Systems — Applications and Theory Vol. 16

COMPUTATIONAL INTELLIGENCE IN SOFTWARE ENGINEERING

W Pedrycz

University of Alberta

J F Peters

University of Manitoba

World Scientific

Singapore • New Jersey • London • Hong Kong

Published by

World Scientific Publishing Co. Pte. Ltd.

P O Box 128, Farrer Road, Singapore 912805

USA office: Suite 1B, 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.

COMPUTATIONAL INTELLIGENCE IN SOFTWARE ENGINEERING

Advances in Fuzzy Systems — Applications and Theory, Vol. 16

Copyright © 1998 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 981-02-3503-8

This book is printed on acid-free paper.

Printed in Singapore by Uto-Print

CONTENTS

PREFACE.....	ix
NEUROCOMPUTING IN SOFTWARE ENGINEERING.....	1
On the Promise of Neural Networks to Support Software Testing.....	3
<i>A. von Mayrhauser, C. W. Anderson, T. Chen, R. Mraz, C. A. Gideon</i>	
Neural Networks for Software Quality Prediction.....	33
<i>T. M. Khoshgoftaar, E. B. Allen</i>	
Self-Organizing Maps and Software Reuse.....	65
<i>D. Merkl</i>	
The Case for an Inductive Computing Science.....	97
<i>D. Partridge</i>	
EVOLUTIONARY COMPUTING IN SOFTWARE ENGINEERING.....	125
Automatic Creation of Computer Programs for Designing Electrical Circuits	
Using Genetic Programming.....	127
<i>J. R. Koza, F. H. Bennett III, D. Andre, M. A. Keane</i>	

The Computer ZOO - Evolution in a Box	151
---	-----

J. Skipper

An Application of Genetic Programming to Software Quality Prediction.....	175
---	-----

T. M. Khoshogoftaar, M. P. Evett, E. B. Allen, P.D. Chien

GRANULAR COMPUTING IN SOFTWARE ENGINEERING.....	197
--	------------

Object Network: A Formal Model to Develop Intelligent Systems.....	199
--	-----

R.R. Gudwin, F.A.C. Gomide

Approximate Reasoning about Complex Objects in Distributed Systems:	
---	--

Rough Mereological Formalization.....	237
---------------------------------------	-----

L. Polkowski, A. Skowron

FOOD: Towards Fuzzy Object-Oriented Systems Design	269
--	-----

W. Pedrycz, Z. A. Sosnowski

Knowledge-Based Techniques for Software Quality Management.....	295
---	-----

C. Ebert, E. Baisch

Fuzzy Logic Techniques for Software Metric Models of Development Effort.....	321
--	-----

A. R. Gray, S.G. MacDonell

Time-Constrained Software Cost Control System: Concepts and Roughly Fuzzy	
---	--

Petri Net Model.....	339
----------------------	-----

J. F. Peters, S. Ramanna

Expert System Design to Control the Information Provided to Pilots.....	371
---	-----

M. S. McCoy, R. R. Levary

Approximating Block Accesses in Database Systems.....	391
---	-----

L. Cui, A. Kandel

NON-CARTESIAN APPROACH TO SOFTWARE

ENGINEERING.....	427
-------------------------	------------

Impact of Non-Cartesianism on Software Engineering.....	429
---	-----

T. Gomi

BIBLIOGRAPHY AND WWW RESOURCES.....	463
--	------------

Computational Intelligence and Software Engineering: A Bibliography and

WWW Resources.....	465
--------------------	-----

E. Pedrycz

INDEX.....	483
-------------------	------------

This page is intentionally left blank

PREFACE

With the rapidly growing demand for new software systems having increasing complexity and size, it is not amazing that the software crisis recognized several decades ago is still with us. As a discipline, software engineering is striving to meet the continuously growing challenges in the area by establishing sound and complete methodologies as well as crystallizing in more specific and fully reliable products (Pedrycz and Peters, 1998). While some extremely useful paradigms have been announced and promoted including software reusability and portability, formal verification methods, to name a few, their transfer into everyday projects is still very slow and limited. Similarly, a number of software measures whose use could (potentially) lead to more reliable products have never been fully exploited in practice.

The recently emerging area of Computational Intelligence (CI) provides a software designer and software professional with a unique conceptual and algorithmic opportunity by taking advantage of the currently developed and already highly mature technologies of fuzzy sets, rough sets, neural networks and evolutionary computation including genetic algorithms. In general, the main objective of CI is to establish a highly coherent design and analysis environment (Bezdek, 1992; Pedrycz, 1997; Peters and Pedrycz, 1998; Pedrycz, Peters and Ramanna, 1998; Peters and Ramanna, 1998a) through a series of synergistic links that give rise to neurofuzzy systems, evolutionary neural networks, fuzzy genetic schemes, granular rough decision systems, and many others in the context of software engineering (see Fig. 1). It is essential to elaborate in brief on these main conceptual and functional pillars of CI.

CI technologies have been used separately and in various combinations to solve problems stemming from increasingly complex forms of software systems description and analysis. This chapter gives a brief introduction to four CI technologies with natural ties to software engineering: fuzzy, rough, neural and evolutionary computing. The intent in introducing CI technologies in software engineering is to address the needs arising during various phases of software development and analysis which involves imprecise measurement and uncertain information.

Examples of the applications of CI technologies in software engineering are summarized

in Fig. 1. Fuzzy sets have been used in object-oriented design (Pedrycz and Sosnowski, 1998) and software quality management (Ebert, 1993). Neural networks have been used in connection with software testing (Mayrhauser and others, 1995; Mayrhauser and others, 1998) and software reuse (Merk1, 1993; Merk1, 1998). Rough sets have been used in assessing software quality (Peters and Ramanna, 1998a; Peters and Ramanna, 1998b) and estimating software cost (Peters and Ramanna, 1996). Evolutionary computing techniques have been used in the study of the evolution of competing programs (Koza, 1993). The micro-cubes along the surface of the CI sphere in Fig. 1 are suggestive of points of convergence of these methodologies that have been used in software engineering. Examples of the combined used of CI technologies can be found in assessing software quality (Peters and Ramanna, 1998b) and in describing and analyzing intelligent systems (Gudwin and Gomide, 1998). Other applications of CI technologies in software engineering can be found in (Pedrycz and Peters, 1997; Pedrycz, Peters, and Ramanna, 1998).

Fig. 1 CI in SE Sphere

CI technologies dovetail with the reality of software system design. For example, an inherent factor of software complexity calls for various approaches capable of handling uncertainty, linguistic evidence, and experimental data. A variety of illustrative examples of the application of CI in software engineering are contained in this volume.

Neural networks (Minsky and Pappert, 1969; Rumelhart and McClelland, 1986) offer a powerful and distributed computing architecture equipped with significant learning abilities. They help represent highly nonlinear and multivariable relationships that can be learned from experimental data. Neural networks have become instrumental in carrying out a vast number of different prediction tasks.

Granular computing forms a key methodology for representing and processing linguistic or, in general, non-numeric information. It supports a diversity of mechanisms of knowledge representation focusing on a relevant selection of information granules and delivering suitable mechanisms aimed at their processing. Fuzzy sets (Zadeh, 1973) exploit imprecision in an attempt to make system's or process' complexity manageable. Rough sets (Pawlak, 1991) help exploit the notions of discernability, indiscernability, and approximate reasoning.

The concept of evolutionary computation (Fogel et al., 1966; Holland, 1975; Koza, 19xx) embraces genetic algorithms, evolutionary computation and evolutionary strategies. All of them are biologically - inspired methodologies exploiting an idea of population-oriented development that is aimed at global system optimization.

The book is intended to become a comprehensive and fully updated source on CI portrayed as a new and useful paradigm in software engineering. It is aimed at practitioners in the area addressing the needs of those fully familiar with software engineering and CI. No prerequisites are necessary. The book addresses the needs of undergraduate students in a number of courses in Computer Science and Computer Engineering curricula in the area of software analysis and design. There are a number of graduate offerings that could highly benefit from the book; these pertain to more advanced aspects of Software Engineering. The book could be helpful in courses devoted to new innovative aspects of computing and software design. The readers interested in the development of advanced software layers for the Internet can find the material stimulating.

The contributions in this volume come with a number of carefully arranged exercises that help the reader reflect over the material and gain a better insight into the essence of the discussed approaches. This volume is divided into three sections. This taxonomy is driven by the main technologies of CI. Part I is devoted to neurocomputing and SE.

Part II deals with evolutionary methods in the setting of software synthesis and analysis. In Part III, we concentrate on the synergy between software engineering and granular computing represented here mainly by fuzzy sets and rough sets. The assumed organization of the material carries an unavoidable level of ambiguity. Simply, some contributions span across a number of areas embracing granular models, learning abilities implied by neural mechanisms and evolutionary behavior associated with genetic algorithms. Nevertheless, by and large, we have achieved a reasonably high level of coherency of classification resulting therein.

We are confident that in the coming years the ideas of CI will find a solid place in the realm of Software Engineering as an essential and powerful platform of synthesis and analysis. We are highly grateful to the authors for their generous contributions that are indisputably located at the frontiers of this rapidly progressing area. Our thanks go to the Natural Sciences and Engineering Council of Canada (NSERC) which through operating, strategic as well as industry-university grants generously supported this project. The professional and on-time help of all folks from the World Scientific was also instrumental in assuring a smooth progress of the project. We hope that the readers will find this volume thought-provoking, inspiring, and intellectually refreshing.

References

- J. C. Bezdek, On the relationship between neural networks, pattern recognition and intelligence, *J. Approximate Reasoning*, 6, 1992, 85-107.
- C. Ebert, An approach to fuzzy data analysis for software quality control, *Proc. 1st European Congress on Fuzzy and Intelligent Technologies, EUFIT'93, Aachen, September 7-10, 1993*, pp. 1156-1161.
- R.R.Gudwin and F.A.C. Gomide, Object network: A formal model to develop intelligent systems. In W. Pedrczy and J.F. Peters (Eds.), *Computational Intelligence in Software Engineering*. Singapore, World Scientific, 1998.
- J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, The MIT Press, 1993.
- D. Merkl, Structuring software for reuse--The case of self-organizing maps. In *Proc. of the Int. Joint Conf. on Neural Networks (IJCNN'93), Nagoya, Japan, 1993*.

D. Merkl, Self-organizing maps and software reuse. In W. Pedrycz and J.F. Peters (Eds.), *Computational Intelligence in Software Engineering*. Singapore, World Scientific, 1998.

A. von Mayrhauser, Ch. Anderson, R.Mraz, Using a neural network to predict test case effectiveness, *Proc. IEEE Aerospace Applications Conf.*, Snowmass, CO, Feb. 1995.

A. von Mayrhauser, C.W. Anderson, T. Chen, R.Mraz, C.A. Gideon, On the promise of neural networks to support software testing. In W. Pedrczy and J.F. Peters (Eds.), *Computational Intelligence in Software Engineering*. Singapore, World Scientific, 1998.

W. Pedrycz, J.F. Peters, "Computational intelligence in software engineering", *IEEE Canadian Conf. on Electrical & Computer Engineering (CCECE'97)*, May, 1997, pp. 253-257.

W. Pedrycz, J.F. Peters, S. Ramanna, Design of a Software Quality Decision System: A Computational Intelligence Approach, *IEEE Canadian Conf. on Electrical & Computer Engineering (CCECE'98)*, May, 1998a [to appear].

W. Pedrycz, J.F. Peters, *Software Engineering: An Engineering Approach*, J. Wiley, N. York, 1998 (to appear)

W. Pedrycz, *Computational Intelligence: An Introduction*. CRC Press, Boca Raton, FL, 1997

J.F. Peters, W. Pedrycz, Computational Intelligence. In Webster (Ed.), *Encyclopedia of Electronic and Electrical Engineering*. New York, John Wiley [to appear].

J.F. Peters and S. Ramanna, A rough sets approach to assessing software quality: Concepts and rough Petri net models. In: *Rough-Fuzzy Hybridization: New Trends in Decision-Making*, edited by L. Polkowski and A. Skowron. Physica Verlag, a division of Springer Verlag [to appear].

J.F. Peters and S. Ramanna, Software Deployability Decision System Framework: A Rough Sets Approach, *Proc. IPMU'98*, Paris, France, July 1998.

J.F. Peters and S. Ramanna, Application of Choquet Integral in Software Cost Estimation. *IEEE Int. Conf. on Fuzzy Systems*, New Orleans, 1996, 862-866.