ELSEVIER

Information and Software Technology 45 (2003) 371–372

INFORMATION AND SOFTWARE TECHNOLOGY

www.elsevier.com/locate/infsof

Editorial

# Introduction to software engineering with computational intelligence

The increasing demand for complex applications in diversified areas imposes a great challenge on developing software systems in order to deal with imprecise and uncertain information. One of the foci of the recent developments in software engineering is the investigation of computational intelligence (CI) for software engineering to address the ever-increasing complexity and size of software systems, and the imperfect information inherited in nature. These treatments enable the extension of CI to various phases in software life cycle along three dimensions:

- Managing fuzziness resided in the requirements, including the formulation of imprecise requirements and the trade-off analysis for conflicting requirements.
- Coping with fuzzy objects and imprecise knowledge, including the modeling of objects and their relationships.
- Handling uncertainty encountered in quality prediction, including the utilization of learning mechanisms to develop general predication models for spotting the most trouble-prone modules early in the software life cycle.

In this volume, we are featuring six papers devoted to the extension of CI to software engineering as a special issue.

Z. Xu and T.M. Khoshgoftaar developed a fuzzy expert system to support independent assessments of projects during the very early phases of the software life cycle. This approach illustrates how a fuzzy expert system can infer useful results by using the limited facts about a current project, and rules about software development. A fuzzy expert system for operational risk assessment is built based upon the NASA standard for early risk assessments, and two previously developed projects from NASA are applied by this system.

S.G. MacDonell takes into account the applicability of fuzzy logic modeling methods to the task of software source code sizing based on a previously published data set. The author uses simple clustering and rule extraction methods to generate first-cut fuzzy models for two samples from a set of 4GL project records and compares the accuracy of these models to those that are achieved via regression-based prediction. This paper concludes that, particularly with refinement using data and knowledge, fuzzy predictive models can outperform their traditional regression-based counterparts.

M. Reformat, W. Pedrycz, and N.J. Pizzi advocate an approach to supporting quality assessment of individual objects. This approach exploits techniques of CI that are treated as a consortium of granular computing, neural networks, and evolutionary techniques. Self-organizing maps and evolutionary-based decision trees are used to gain a better insight into the software data and to support a process of classification of software objects. Genetic classifiers serve as “filters” for software objects. Using these classifiers, a system manager can predict quality of software objects and identify low quality objects for review and possible revision.

C. Matthews provides a fuzzy logic toolkit for the formal specification language Z to permit the incorporation of fuzzy concepts into the language while retaining the precision of Z specifications. The toolkit provides the necessary operators and modifiers for the definition and manipulation of fuzzy sets and relations. The focus of this paper is on the specification of the rule base and the operations necessary for fuzzy inferencing, and on the illustration of how the fuzzy logic toolkit can be used in the specification of simple fuzzy expert systems.

N. Marín, J.M. Medina, O. Pons, D. Sánchez, and M.A. Vila offer a set of operators that are useful in comparing objects in a fuzzy environment. For the aim to deal with imprecise and imperfect objects, they introduce a generalized resemblance degree between two fuzzy sets of imprecise objects and a generalized resemblance degree for comparing complex fuzzy objects within a given class.

J. Lee and Y.Y. Fanjiang propose a new approach to (1) defining fuzzy object-oriented modeling (FOOM) schema based on the XML schema, and incorporating the notion of stereotypes to facilitate the modeling of imprecise requirements; and (2) transforming the FOOM schema into a set of APIs through the use of

0950-5849/03/$ - see front matter 
doi:10.1016/S0950-5849(03)00009-0

schema graph as an intermediate representation for content validation and data access in an XML document automatically.

I would like to express my sincere gratitude to everyone who has contributed to this special issue including the authors, the reviewers, and the Co-Editor-in-Chief Dr Michael Dyer.

Jonathan Lee*
*Department of Computer Science and Information Engineering, National Central University, Chungli 32054, Taiwan*
*E-mail address:* yjlee@selab.csie.ncu.edu.tw

* Tel.: +886-3-42271514510; fax: +886-3-4222681.