Chapter 11

Cognitive Complexity Measures: An Analysis

Sanjay Misra
Federal University of Technology, Nigeria

ABSTRACT

Cognitive informatics (CI), a multidisciplinary area of research tries to solve the common problems in the field of informatics, computer science, software engineering, mathematics, cognitive science, neurobiology, psychology, and physiology. Measurement in software engineering is also a core issue which is still striving for its standardization process. In recent years, several cognitive complexity measures based on CI have been proposed. However, each of them has their own advantages and disadvantages. This chapter presents a critical review on existing cognitive complexity measures. Furthermore, a comparative study based on some selected attributes has been presented.

INTRODUCTION

Software metrics play an important role in the software development process, since; they assist the software developer in assuring the software quality. Developers use these metrics in the entire life cycle of software development to better understand and assess the quality of engineered products or systems that they built. On the other hand, it is a common observation that it is not possible to develop the absolute measure (Fenton, 1994). Instead, software engineers attempt to derive a set of indirect measures that lead to metrics that provide an indication of quality of some representation of software. The quality objectives may be listed as performance, reliability, availability and maintainability (Somerville, 2001) and are closely related to software complexity.

IEEE (1990) defines software complexity as "the degree to which a system or component has

DOI: 10.4018/978-1-60960-215-4.ch011

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

a design or implementation that is difficult to understand and verify". Further, there are two categories of software complexities: computational and psychological (Fenton, 1997). Computational complexity is related to algorithm complexity and evaluates time and memory needed to execute a program. The second is psychological complexity also called as cognitive complexity is concerned to evaluate the human effort needed to perform a software task. This definition is also related to understandability of the software and task and evaluates the difficulty experienced in comprehending and/or performing such a task. Further, there are several definitions of the cognitive complexity. Henderson-Sellers (1996) define the cognitive complexity as 'The cognitive complexity of software refers to those characteristics of software that affect the level of resources used by a person performing a given task on it.' Fenton (1997) defines cognitive complexity as it measures the effort required to understand the software. Zuse (1998) definition of complexity also represents the notion of cognitive complexity, which states that software complexity is the difficulty to maintain, change and understand software. Here, it is worth mentioning that metrics and measures are often used as synonyms terms in software engineering. It is because of the fact that both terms have approximately similar definitions. Pressman (2001) explains the measure in software engineering context as 'a measure provides a quantitative indication of the extent, amount, dimension, capacity, or size of some attributes of a product or process'. A metric is defined by IEEE as 'a quantitative measure of the degree to which a system, component, or process possesses a given attribute".

Cognitive informatics (CI), a multidisciplinary area of research is emerging. It includes the researches in the field of informatics, computer science, software engineering, mathematics, cognitive science, neurobiology, psychology, and physiology (Wang, 2002, 2004, 2005, 2006, 2007, 2009). The importance of the researches in CI is due to the fact that, it tries to solve the common problems of two related area in a bi-directional and multidisciplinary approach (Wang, 2004). CI uses the computing technique to solve the problem of cognitive science, neurobiology, psychology, and physiology and on the other hand the theories of cognitive science, neurobiology, psychology, and physiology are used to investigate the issues and their solution in informatics, computing, and software engineering. For examples, the measurement in software engineering is a major issue which is still not mature and needs a lot of efforts to standardize it (i.e. the measurement techniques for software engineering). In last few years numbers of researchers have tried to solve these problems by combining the principles of cognitive science and (measurement in) software engineering. The numbers of proposals of cognitive complexity measures (Shao & Wang, 2003; Misra, 2006, 2007, 2010; Kushvaha & Misra, 2006; Auprasert & Limpiyakorn, 2008, 2009) are the results of these efforts.

Cognitive Complexity refers to the human effort needed to perform a task or the difficulty experienced in understanding the code or the information packed in it (Misra & Kushvaha, 2006). Understandability of the code is known as program comprehension and is a cognitive process and related to cognitive complexity. In other words, the cognitive complexity is the mental burden on the user who deals with the code, for example the developer, tester and maintenance staff. Cognitive complexity provides valuable information for the design of systems. High cognitive complexity indicates poor design, which sometimes can be unmanageable (Briand, Bunse & Daly, 2001). In such cases, the maintenance effort increases drastically. In this respect, cognitive complexities are important in evaluating the performance of the system; they refer to those characteristics of software which affect the level of resources used by a person performing a given task on it (Zuse, 1998). The system with reduced cognitive complexity will not only improve the quality of

15 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/cognitive-complexity-measures/51976

Related Content

Plant Leaf Disease Detection Using CNN Algorithm

Deepalakshmi P., Prudhvi Krishna T., Siri Chandana S., Lavanya K.and Parvathaneni Naga Srinivasu (2021). *International Journal of Information System Modeling and Design (pp. 1-21).*

www.irma-international.org/article/plant-leaf-disease-detection-using-cnn-algorithm/273224

Towards a Theoretical “Cybernetic” Framework: Discovering the Pedagogical Value of the Virtual World “Second Life”

Pellas Nikolaos (2013). *Software Development Techniques for Constructive Information Systems Design (pp. 128-181).*

www.irma-international.org/chapter/towards-theoretical-cybernetic-framework/75745

Assessment of BAR: Breakdown Agent Replacement Algorithm for SCRAM

Shivashish Jaishy, Yoshiki Fukushige, Nobuhiro Ito, Kazunori Iwataand Yoshinobu Kawabe (2017). *International Journal of Software Innovation (pp. 1-17).*

www.irma-international.org/article/assessment-of-bar/182533

Combinatorial Testing

Renée C. Bryce, Yu Lei, D. Richard Kuhnand Raghu Kacker (2010). *Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization (pp. 196-208).*

www.irma-international.org/chapter/combinatorial-testing/37033

Adaptive Computation Paradigm in Knowledge Representation: Traditional and Emerging Applications

Marina L. Gavrilova (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications (pp. 3258-3271).*

www.irma-international.org/chapter/adaptive-computation-paradigm-knowledge-representation/29560