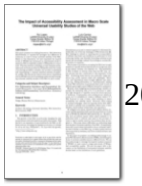
A multi-paradigm complexity metric (MCM)

Authors: Sanjay Misra Faculty of Engineering, Department of Computer Engineering, Atilim University, Ankara, Turkey

Ibrahim Akman Faculty of Engineering, Department of Computer Engineering, Atilim University, Ankara, Turkey

Ferid Cafer Faculty of Engineering, Department of Computer Engineering, Atilim University, Ankara, Turkey



2011 Article

Published in:
· Proceeding
ICCSA'11 Proceedings of the 2011 international conference on Computational science and Its applications - Volume Part V
Pages 342-354
Springer-Verlag Berlin, Heidelberg

table of contents ISBN: 978-3-642-21933-7

Huge amount of researches and software metrics have been proposed for procedural and object-oriented languages. However, there are only few metrics available in the literature related with multi-paradigm programming languages. In this paper, we propose a metric to evaluate the code written in multi-paradigm language. Our proposed metric can be used for most of the programming paradigms, including both procedural and object-oriented languages.

Keywords: Software complexity, complexity melrics, Python, software development.

1 Introduction

Software qual ity has been raising demand for decades due to its complexity. Software
development processes are treated as a complex task, therefore to control its complexity
and to maintain the quality of software is a challenging job. A software product should carry several quality attributes, such as correctness, relialbility, efficiency, integrity,
usability, maintainabi li ty, testability, flexibili ty, portabi liLy, reusability, and interoperability [1]. According to Somerville [2] the most necessary software quality
anribute is maintainability and, to be able to maintain a software system efficiently, the codes should be understandable for the developers. This means, to achieve high quality code, reduction of code complexity is essential. Metrics are indicators of complexity
and can be used to estimate software quality. There are a number of metrics each focusing on different complexity factors [3].

One may find the hundreds of metrics in the li terature for evaluating the software quality. These metrics [4-7 , 18-25] are of several types and for differe nt purposes. Somerville [2] categorizes metrics as control and predictor metrics. Control metrics are related to software processes, whereas predictor metrics are .associated with software
products. Control metrics estimate effort, time and defects and predictor metrics assess the number of attributes and structures in a code. The literature provides several
metrics for procedural and 00 paradigms. However, the re are very few metrics

for multi paradigm languages [26-27]. This constitutes our motivation and the present
paper proposes a metric for multi paradigm la nguages, which combines the features of
procedural and 00 languages.
Generally, code complexity depends on its external and internal c haracteristics which can be c haracterized by functionality and efficiency respectively. In order to
8. Murgantc et al. (Eds.}: tCCSA 20 ll. Part V. LNCS 6786. pp. 342-354.2011.