Computers and Electrical Engineering 000 (2017) 1–16

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Software reusability metrics estimation: Algorithms, models and optimization techniques☆

Neelamdhab Padhy[a,*], R.P. Singh[b], Suresh Chandra Satapathy[c]

[a] Research Scholar, Department of Computer Science and Engineering, Sri Satya Sai University of Technology and Medical Science (SSSUTM), India
[b] Professor and Head of PVP institute, Vijayawada, India
[c] Vice-Chancellor of SSSUTM, Sehore, India

ARTICLE INFO

Article history:
Received 30 July 2017
Revised 19 November 2017
Accepted 20 November 2017
Available online xxx

Keywords:
Software reusability metrics
Software metrics
Aging-resilient
Software reusability prediction
Software metrics algorithms

ABSTRACT

Objective: In this paper, the proposed model is intended to employ a novel evolutionary computing-based artificial intelligence or machine learning scheme for regression tests to be used for reusability estimation. Such enhancement can lead to accurate reusability pattern estimation, which can be effective for optimal software design purposes. This model is popularly called an aging-resilient software reusability forecast representation. The proposed system employs predominant object-oriented software metrics, such as Chidamber and Kemerer's metrics to examine reusability. Here, cumulative metrics, object-oriented metrics, McCabe's metrics, cohesion and a coupling-based reusability assessment model have been proposed which could be of paramount significance in software design optimization. In this paper, software metrics algorithms and their primary constructions have been developed for estimating the metrics from the UML/class diagrams. It is feasible to derive an efficient and robust reusability prediction model for web-service products using object-oriented metrics. Here, it was also found that OO-CK metrics, particularly complexity, cohesion and coupling-related metrics can be helpful in predicting reusability in web-service software products. Considering the above-mentioned key contributions, it can be stated that the proposed research could be of paramount significance in next-generation software computation systems, primarily for software component reusability, reliability, survivability, aging prediction and stability, and for software excellence assurance purposes.

1. Introduction

Software can be said to be one of the more significant inventions affecting human society, as it includes varied application horizons such as science and technology, business, finance, medicine, defense, space technologies and aircraft, together with an array of industrial utilities and various security systems. Identifying software as an un-substitutable entity in modern human life should not be surprising, because it has enabled and facilitated comfort in every functional activity for humans in the present and can be expected to do so in the future. All these aspects demonstrate the significance of software in human life and in science and technology. Reusing existing or conveniently available software components to reduce time

☆ Reviews processed and recommended for publication to the editor-in-chief by guest editor Dr. A. P. Pandian.

* Corresponding author.neelamadhabphd@gmail.com
E-mail addresses: neelamadhabphd@gmail.com (N. Padhy), prof.r.p.singh@gmail.com (R.P. Singh), sureshsatapathy@ieee.org (S.C. Satapathy).

https://doi.org/10.1016/j.compeleceng.2017.11.022
0045-7906/

and costs for computation and human resources is called software reusability. Obtainable code is frequently used in new projects. This is the main attribute of the maximum possible use characteristic for developing software components. There are various ways to obtain reusability benefits such as cost, reliability and better-quality service [1]. At present, software module-based software design (SMBSD) is of additional interest to researchers. This provides a new measurement style for developing components in the industry. The reason behind it is low cost and quality software design. These days, the rising cost of software-based computations has prompted academia and corporate industries to develop certain optimal software design solutions that might ensure cost-efficient software systems without compromising reliability, scalability, stability and overall quality. Achieving software quality while ensuring minimum cost of design could strengthen the software industry, enabling it to remain relevant in the market and to achieve a higher market share. Additionally, ensuring reliability (survivability, aging-resilient processing) could be of paramount significance for both the users and the industries producing software as a part of various utilities. In general, a software system encompasses various modules and packages, sometime called components, which encapsulate significant portions of related data and functions. In practice, all system processes are placed into distinct classes so as to maintain the semantic relations between the data and functions in each module or class. Designing or developing software systems using already available modules or classes is known as reusability, and can have major significance. In a software engineering approach, software metrics require more attention in order to improve the quality of the product. Developers mainly focus on quality not quantity. Metrics are primarily used for the measurement of a product or a component. Developing an effective software system is a challenging task. Reusable components can be developed via existing source code. Once they are developed, they can be used further in different classes. Achieving full reuse is part of the fruitfulness of software systems. This provides lower cost and better performance. The major motivation for developing such components is the time factor, i.e. the time within which a stipulated stage of development of the new component can be achieved. These components significantly reduce software development cost and time. The subject of software engineering is the dominant area in software development where important products can be carefully considered. In the current scenario, developers consider reusability to be a high-quality attribute. Overall, the development cost will be reduced by 43% by implementing the software reusability approach. This includes processes whereby new software can be built using existing software. The software design can be considered as an approach making use of abstraction, selection, instantiation and integration. Research into software reusability will be helpful to the industry and to the software reusability community. Chidamber and Kemerer [2] proposed six metrics, called the Chidamber and Kemerer software metrics.

This paper has been organized as follows. Section 2 describes the literature survey and discusses the estimation techniques of software reusability metrics. This section also describes software aging. Here, aging means the identification of software faults. In this part of the paper, the reasons behind aging and its classification are explained, together with how different researchers present the reasons for software aging. Sections 3 and 4 discuss software metrics and reusability. In these sections, in-depth explanations are given of CK-related metrics, reusability features, inheritance, classes and function-related metrics described by different researchers. Section 5 highlights the main research question and its required solutions. Sections 6 and 7 introduce the research gap and objectives, including classification of the general and specific objectives of the research paper. Section 8 presents the research methodology. This methodology proposes reusability estimation techniques such as EC-AI-based regression analysis for reusability estimation and aging- and survivability-related reusability optimization models. Section 9 describes the research contribution. Sections 10 and 11 provide the software reusability metrics algorithms and formal structures. These algorithms are proposed based on object-oriented software code.

1.1. Software reuse techniques

The following techniques have been adopted to achieve software reusability. These techniques help to categorize the reusability factor and support the choice of parameters and metrics concerned with reusability. They include: evolutionary computing-based ANNs for software reusability prediction, support vector machines, neural networks, genetic algorithms and fuzzy logic. The inspiration for this paper was to recognize and estimate the current state of the art for software reusability metrics estimation, including requirements, approaches, models and optimization techniques. The intention of this review is to benefit practitioners and researchers who wish to assess the advantages and disadvantages of the requirements of various reuse approaches.

1.2. Research highlights

The primary contribution is that the proposed AGA-ANN-based reusability prediction model is better than any other available approaches, and therefore can be applied for real-time application purposes. Other highlights are:

- √ Identification of the reusability factor which influences software code.
- √ Reusability rank estimation. A proposed equation has been developed which will estimate the rank.
- √ Prediction of the percentage of code reused in new components. A proposed equation has been developed for identifying the percentage of code that will be reused.
- √ Measurement of the reusability in the class. An equation has been constructed to measure the reusability in the class or component.
- √ Estimation of the software reusability. A software reusability estimation model has been proposed and an aging- and survivability-related reusability optimization model has been developed.

Please cite this article as: N. Padhy et al., Software reusability metrics estimation: Algorithms, models and optimization techniques, Computers and Electrical Engineering (2017), https://doi.org/10.1016/j.compeleceng.2017.11.022