UDC 004.415.5

A. ANDRASHOV, A. GORDEYEV, V. KHARCHENKO, V. SKLYAR

National Aerospace University Named After M.E. Zhukovsky, Ukraine

THE STATIC ANALYSIS OF A PROGRAM CODE PROCEDURE BASED ON METRICS PROFILING

The static analysis of a program code procedure is developed. The analysis and classification of software quality estimation metrics is conducted. For a quantitative estimation of a program code quality, generic metrics' index is proposed. The tool of static analysis process support is developed.

software quality assessment, metrics, static analysis, tool

Introduction

With increase in demand of the software using, the risks connected with refusals and failures which reason are defects of the software increases. Insufficient software quality, as a rule, is consequence of insufficient quality of the processes of its development, testing and verification. This fact causes an urgency of the scientific researches, devoted to development and improvement of the software quality estimation methods.

Software quality, as a rule, is defined by the quantity of residual defects in it. One of the reasons of defects presence is insufficient quality of a program code, in particular, the complexity of topology (control flow graph) and not standardized programming style [1].

As the analysis result of existing static analysis methods and tools, following lacks have been certain: firstly, heterogeneity and absence of metrics classification; secondly, high cost of the program code static analysis tools; thirdly, existing tools are an integral part of software products development platforms and, as a rule, cannot functionate independently (Cantata ++, AdaTEST).

The static analysis methods are applied to an estimation of a program code quality. *The static analysis* is a process of the software analysis without its direct performance [2]. Existing static analysis methods and tool include metrics. *The metric* is a method of estimation and a scale of measurement [3]. For today exists more than fifty metrics of a code quality estimation, but the majority of them are not a part of the software static analysis tools (Together, Testbed, etc.).

In this connection **the purpose** of given article is the static analysis of a program code procedure and tool of the static analysis process support development.

1. The static analysis of a program code procedure

Procedure structure. Suggested procedure (fig. 1) is intended for software quality estimation by the program code static analysis. The procedure is based on «Together» tool metrics and the metrics which have not entered in «Together» interconnecting. On a first step of metrics interconnecting, it has been decided to generate the generic classification (profile). For this purpose the «Together» metrics and metrics not entering in this tool taxonomy has been constructed. Additive convolution can be applied to metrics calculation.

The «Together» tool metrics analysis. The «Together» tool is applied to support software designing and quality estimation (program code audit) [4]. Given tool is developed by «Borland» company. The «Together» contains set of metrics which are focused on a program's code static estimation.

The metrics used in «Together», are the basic means by using which the project manager and the architect can keep watching on development of the program environment. To estimate software quality, using «Together» tool, one have to define a metrics profile, as there is some overlapping between metrics presents. For example, in one metrics set is not necessarily should be a cyclomatic complexity and a maximum size of operation metrics.

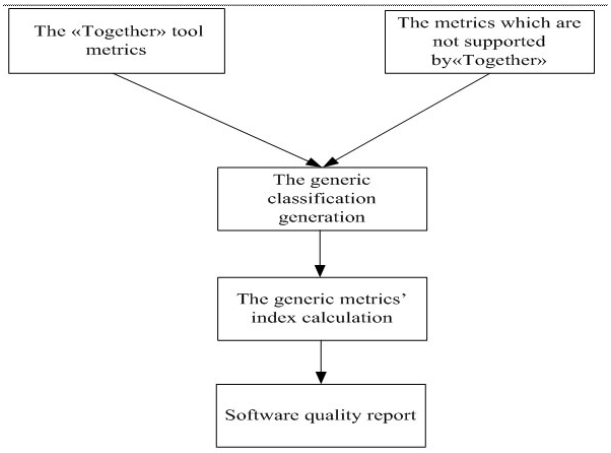

Fig. 1.The static analysis procedure stages

«Together» metrics allow conducting the quantitative analysis of an initial program code by following criteria: basic, complexity, coupling, encapsulation, inheritance, on the basis of a maximum, polymorphism, ratio, audit violations.

The «Together» tool metrics classification (fig. 2) based on conducted analysis has been offered.

The analysis of additional software quality estimation metrics. For generic metrics profile nomenclature expansion the analysis of metrics which are actively used in the modern industry of program engineering has been made. On the grounds of the analysis a number of metrics which application allows sufficiently estimate characteristics of software quality has been allocated [5, 6]. The decision to classify all set of metrics (fig. 3) was accepted.

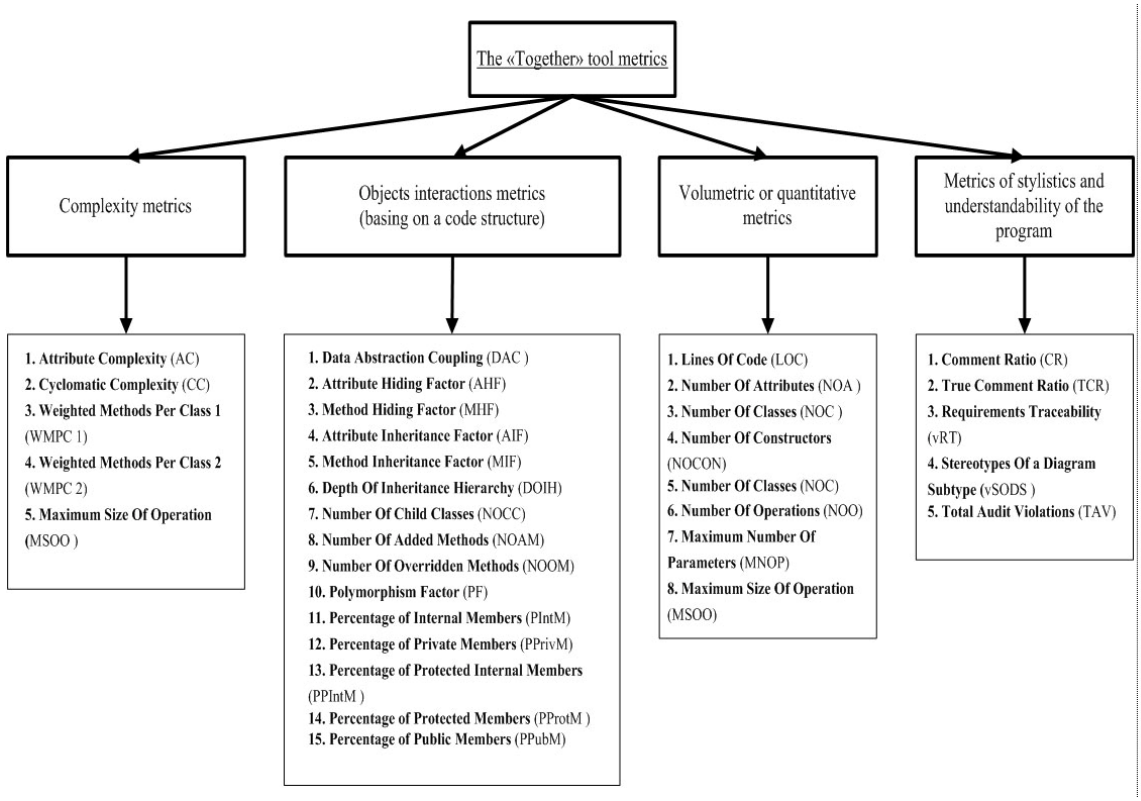

Fig. 2. The «Together» tool metrics classification

Let us consider the metrics classification attributes:

1) the attribute of the testing analysis is based on two directions of testing the static and dynamic analysis (the metrics of the dynamic analysis are not considered);

2) the attribute of a life cycle stage is divided into the following: metrics witch are applied at a stage of coding and designing;

3) the attribute of complexity level is based on complexity of metrics calculation and contents. Complex metrics can include a number of the simple one;

4) under the strategy attribute it is meant two strategies: a white and a black box. I.e. metrics which work with use of internal software structure information and metrics which use the information which is not basing on software structure;.

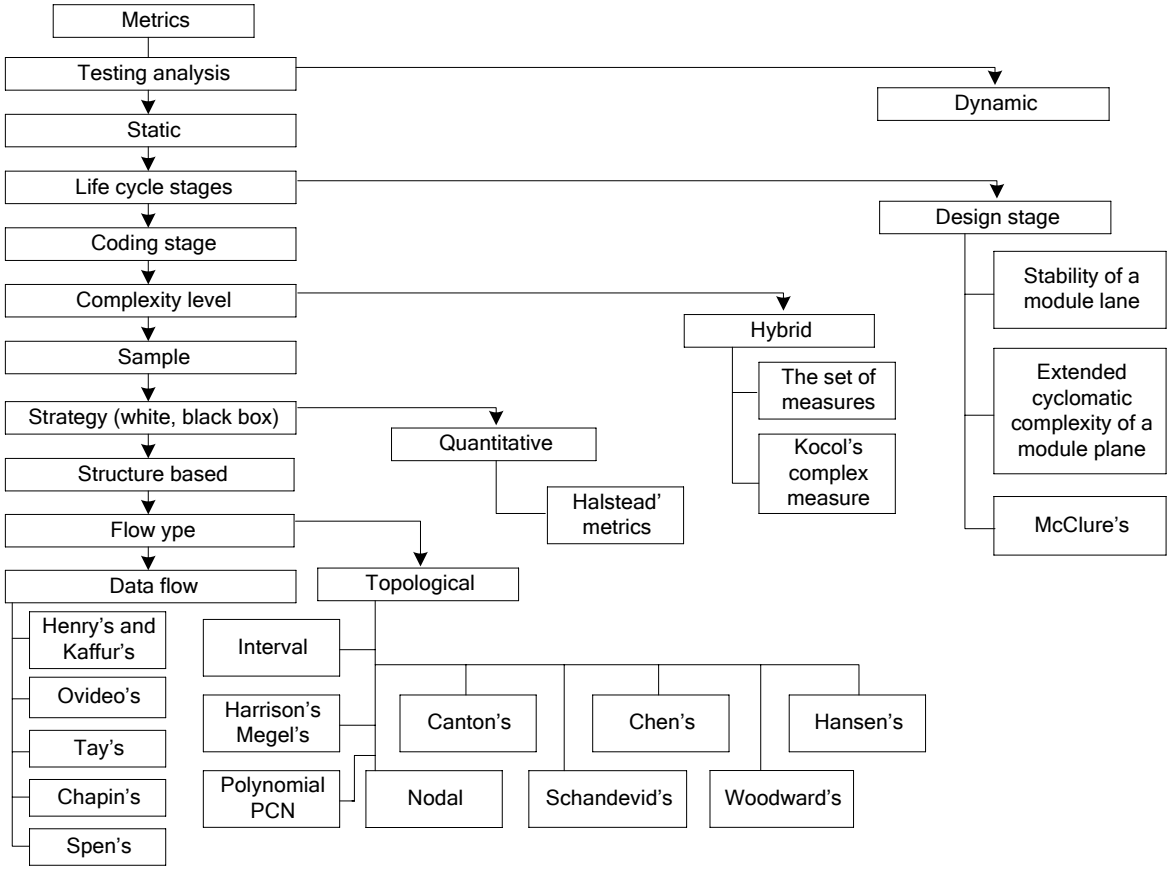

Fig. 3. The additional metrics classification

5) the attribute of a stream means itself the division of metrics on topological and the metrics of a dataflow. The basis of topological metrics is made by software structure. And a basis of the dataflow metrics is definition of software data correctness.

The presented classification included twenty metrics of software quality estimation. Most of the metrics has the full description; however, with regard to some metrics there is no information about their primitives.

The generic metrics profile. On the basis of the conducted analysis and classification of software quality estimation metrics has been drawn a conclusion, that for an all-round estimation of software quality and reliability it is necessary to generate the generic metrics profile (fig. 4). Which will includes the «Together» tool metrics and metrics not entering into it. The basic lack of «Together» tool metrics is absence of a full set of the software complexity metrics.

The generic metrics profile constitutes by it self the facet structure which is formed on the basis of combining of the classification scheme of «Together» tool metrics (fig. 2), transformed into facet structure and the classification scheme of metrics, not supported by «Together» (fig. 3).

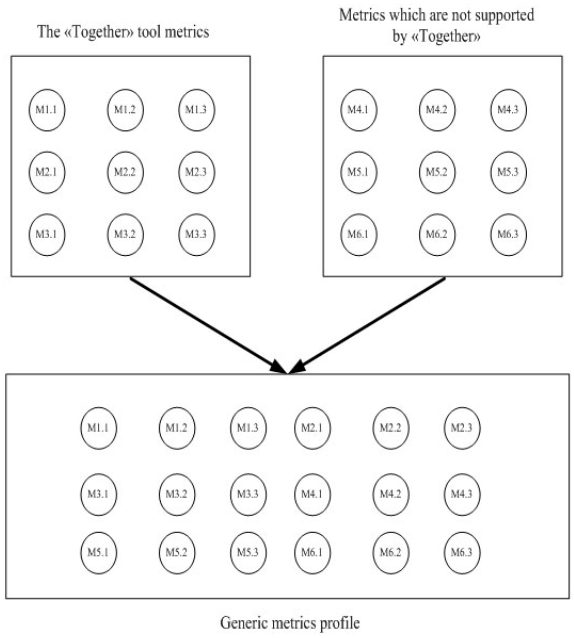

Fig. 4. The generic metrics' profile forming principle

It is necessary to note, that at formation of the generic metrics profile there were no intersections i.e. metrics were not duplicated.

The static analysis of a program code quality index. For a quantitative estimation of the static analysis of a program code quality the following index has been introduced:

The generic metrics' index (GMI) which is calculated using the formula:

$$GMI=M_1*K_1+M_2*K_2+...+M_N*K_N,$$

where M_N – numerical metrics value, K_N – is weighting factor which defines by the experts**,**

$$\sum_{i=1}^{N} K_i = 1.$$

The generic metrics' index is used for a quantitative assessment of metric program's code estimation quality. Due to weight factors, experts have an opportunity to define the importance of each metric at the calculation of an index as a whole.

2. Case study: tool «Togageks»

On the basis of static analysis procedure the tool for static analysis of a program code process support has been developed.

Current tool consists of the following (fig. 5) modules: «Together» module, carrying out the metrics calculation for a program code static analysis; the «metrics calculation» module is necessary for calculation the metrics which are not supported by «Together» module; the «index calculation» module is intended for a calculation of a program code quality estimation, entrance data for which are the metrics values counted by «Together» module and «metrics calculation» module; the «metrics visualization» module is necessary for radial-metric diagram (RMD) forming with the purpose of visual display of metrics values.

The database represents the set of html files containing values of metrics calculations and estimation quality indices.

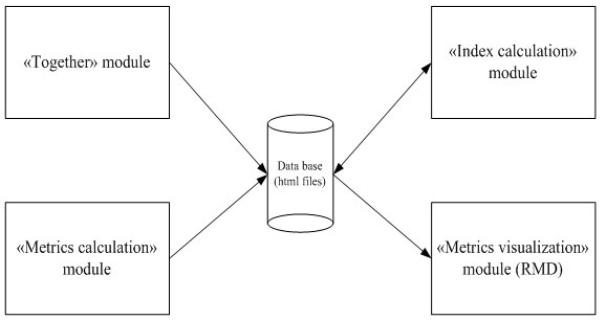

Fig. 5. The «Togageks» tool structure

The «Togageks» tool carries out following functions:

– downloading «Together» html report file from the data base;

– displaying of the given report at the program's form;

– primitive selection and calculation of metrics which are not entering in «Together» tool;

– calculated metrics RMD visualization;

– program code estimation quality index calculation;

– saving metrics and index values into data base.

Program interface is represented as form with a metrics list, downloading from «Together» html report file, metrics visualization window of chosen by user metrics (fig. 6.). There is an opportunity of changing the gradation scale of numerical metrics values at displayed RDM.

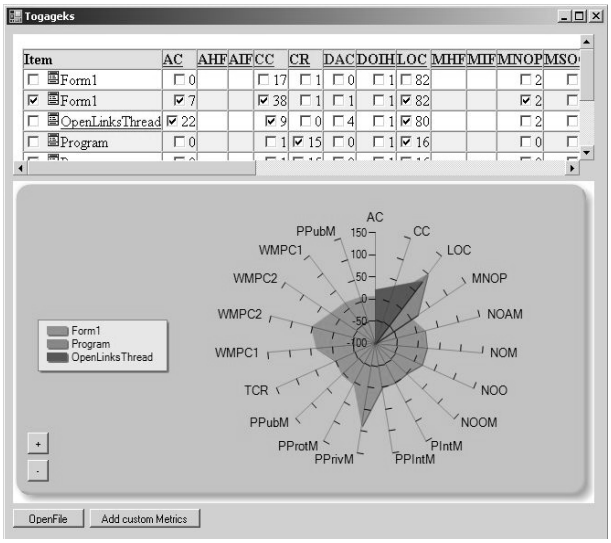

Fig. 6. The «Togageks» tool main form

It is possible to choose a decomposition level of metrics calculation and RMD displays (as for separate classes and modules of tested software, as well for the project as a whole).

The form on (fig.7.) is intended for input the metrics which calculation is not supported by «Together» tool. Metrics primitives are set in an obvious kind in Java-script language.

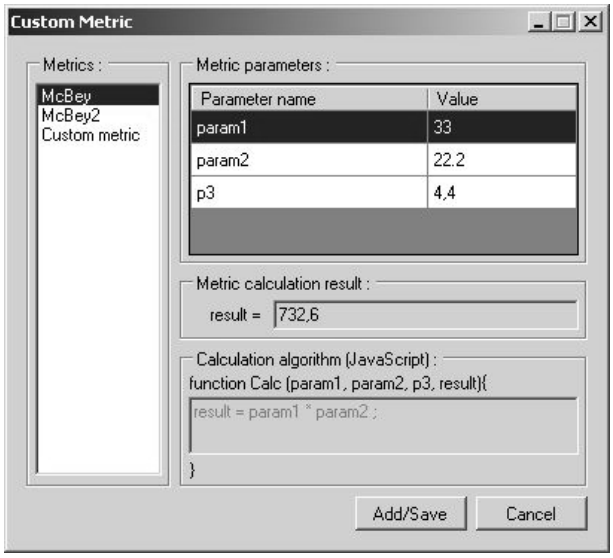

Fig. 7. Metrics' primitive input form

After the primitive assignment the metric value is to be brought in the general table (fig.6.) with an opportunity of the further visualization on RMD. When all the metrics' values are calculated, computing the static analysis quality index. Further all the calculated information in the form of the report is saving in the data base.

Conclusions

The analysis and classification of software quality estimation metrics is conducted. The static analysis procedure, based on the generic metrics profile forming, is introduced. The index of software estimation quality is described. The tool of static analysis process support is developed.

Proposed static analysis of a program code procedure and static analysis process support tool might be used at the software independent verification, expertise and audit.

Further researches are advisably to aim at the development of metrics' profile forming procedure for specific projects.

The «Togageks» tool was used for safety assessment of software of Ukrainian Nuclear Power Plants Instrumentation and Control System. This tool using permits to find some incorrectness in program code that improves quality and safety of critical software.

References

1. Liu K., Zhou S. Yang H., Quality Metrics of Object Oriented Design for Software Development and Redevelopment // Proceedings of the IEEE First Asia-Pacific Conference on Quality Software, 2000.

2. Липаев В.В. Надёжность программных средств. – М.: СИНЕТ, 1998. – 232 с.

3. Кармайкл Э., Хейвуд Д. Быстрая и качественная разработка программного обеспечения. – М.: Издательский дом «Вильямс», 2003. – 391 с.

4. User Guide for Together ControlCenter™. – Borland Software Corporation, 2004. – 922 p.

5. Черноножкин С.К. Меры сложности программ (обзор) // Системная информатика. Вып. 5: Архитектура, формальные и программные модели. – Новосибирск: Наука, 1997. – С. 188-227.

6. Черноножкин С.К. Задачи автоматического построения тестов и статический анализ // Программирование. – № 2. – 2001. – С. 35.

Поступила в редакцию 2.03.2007

Рецензент: д-р техн. наук, проф. Б.М. Конорев, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.