7th International Conference on Applied Informatics
Eger, Hungary, 2007.

Why Code Complexity Metrics Fail on the C++ Standard Template Library[*]

Norbert Pataki[a], Zoltán Porkoláb[a], Edit Csizmás[b]

Dept. of Programming Languages and Compilers,
Fac. of Informatics, Eötvös Loránd University, Budapest
e-mail:{patakino, gsd}@elte.hu

Dept. of Informatics, Fac. of Mechanical Engineering and
Automation Kecskemét College, Kecskemét
e-mail: csizmas.edit@gamf.kefo.hu

Abstract

Since McCabe's cyclometric measure, structural complexity have been playing an important role measuring the complexity of programs. Complexity metrics are used to achieve more maintainable code with the least bugs possible.

C++ Standard Template Library (STL) is the most popular library based on the generic programming paradigm. This paradigm allows implementation of algorithms and containers in an abstract way to ensure the configurability and collaboration of the abstract components. STL is widely used in industrial softwares because STL's appropriate application decreases the complexity of the code significantly.

Many new potential errors arise by the usage of the generic programming paradigm, including invalid iterators, notation of functors, etc.

In this paper we present many complexity inconsistencies in the application of STL that a precise metric must take into account, but the existing measures ignore the characteristics of STL.

Keywords: metrics, generic programming

MSC: 68N30 Mathematical aspects of software engineering

Norbert Pataki
Pázmány Péter sétány 1/c., H-1117 Budapest, Hungary

Zoltán Porkoláb
Pázmány Péter sétány 1/c., H-1117 Budapest, Hungary

Edit Csizmás
Izsáki út 10., H-6000 Kecskemét, Hungary

[*]Supported by GVOP-3.2.2.-2004-07-0005/3.0