Chapter V

Morphology, Processing, and Integrating of Information from Large Source Code Warehouses for Decision Support

Jörg Rech, Fraunhofer Institute for Experimental Software Engineering (IESE), Germany

Abstract

Source code occurs in diverse programming languages with documentation using miscellaneous standards, comments in individual styles, extracted metrics or associated test cases that are hard to exploit through information retrieval or knowledge-discovery techniques. Typically, the information about object-oriented source code for a software system is distributed across several different sources, which makes processing complex. In this
chapter we describe the morphology of object-oriented source code and how we (pre-)process, integrate and use it for knowledge discovery in software engineering in order to support decision-making regarding the refactoring, reengineering and reuse of software systems.

Introduction

Traditionally, databases storing software artifacts were used to store manually classified components from in-house software systems. But as the manual classification of source code is time-consuming and costly, automated techniques for software retrieval and reuse are required to efficiently and effectively process large amounts of source code.

Today, we use code warehouses to store many software systems in different versions for further processing, which are very similar to the data warehouse framework described by Inmon (Inmon, 1996). Operational configuration management systems (CMS), similar to data marts, are tapped to integrate software artifacts into the code warehouse. Typically, these software repositories consist of a vast quantity of different files with interconnected source code and additional information associated with the source code (e.g., documentation). Extraction, transformation and loading processes (ETL) are used to extract the source code from different software repositories and varying languages and formats into the code warehouse. Due to the astonishing success and propagation of open source software (OSS) and large OSS repositories, such as Sourceforge (cf. http://www.sourceforge.net), many CMSs are freely available in different shapes and sizes. By tapping these operational code marts, large amounts of reusable software artifacts in diverse languages, formats and with additional information are available for further reuse, analysis and exploration.

But as code warehouses and single software systems grow larger, it becomes more complicated to decide on changes or enhancements. The complexity and dissimilarity of the source code has to be unified in order to adjust knowledge discovery (KDD) or information retrieval (IR) techniques. Tasks such as (agile) software reuse, refactoring or reengineering are increasingly more complex and intensify the need for decision support in SE (Ruhe, 2003). Techniques from artificial intelligence (AI) are being used in SE (Rech & Althoff, 2004) to support managerial decisions (e.g., where to focus testing effort) as well as design decisions (e.g., how to structure software systems). Several approaches for the discovery of knowledge to support decisions in software reuse and quality improvement have been developed (Rech et al., 2001), but support of decisions for reuse and refactoring based on large code warehouses is still unsatisfactory.
Application of Probabilistic Techniques for the Development of a Prognosis Model of Stroke Using Epidemiological Studies
www.igi-global.com/article/application-of-probabilistic-techniques-for-the-development-of-a-prognosis-model-of-stroke-using-epidemiological-studies/105930?camid=4v1a

A Social-Academic Network Analysis of the EURO Working Group on DSS
www.igi-global.com/chapter/social-academic-network-analysis-euro/66737?camid=4v1a