Abstract

Our ongoing effort, from which we draw the work reported here, is focused on the case of reorganization of legacy software, consisting of millions of line of non-object oriented code, that was never modularized or poorly modularized to begin with. We can think of the problem as reorganization of millions of lines of code residing in thousands of files in hundreds of directories into modules, where each module is formed by grouping a set of entities such as files, functions, data structures and variables into a logically cohesive unit. Furthermore, each module makes itself available to the other modules (and to the rest of the world) through a published API.

References

- S. Sarkar, A. C. Kak, and N. S. Nagaraja, "Metrics for Analyzing Module

- Java 2: The Complete Reference, Fifth Edition by Herbert Schildt
- Stein, Benno; Koppel, Moshe; Stamatatos, Efstathios; “Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection PAN’07”;
- Potthast, Martin; Stein, Benno; Eiselt, Andreas; Barrón-Cedeño, Alberto; Rosso, Paolo (2009), “Overview of the 1st International Competition on Plagiarism Detection”;

Index Terms

Computer Science
Emerging Trends in Technology

Keywords

Application Programming Interface(api)  Modularization function Call Traffic  Non Api metrics