

# Module Metric Signature (MMS) Visualization

Dolores Zage Wayne Zage  
Software Engineering Research Center  
Computer Science Department  
Ball State University  
Muncie, IN 47306 USA  
{dmzage, wmzage}@bsu.edu

## Abstract

We have been developing a tool to enhance the use of metrics during software development. This tool is a metrics visualization environment that graphically depicts the innate structure of software through a module signature along with an element of time. We call the tool *Module Metric Signature (MMS) Visualization*.

## 1. Introduction

An intuitive and comprehensive viewing environment is necessary for metrics to highlight favorable and unfavorable trends for designers, developers, managers, testers and maintainers. However, metrics, even though very useful, can require much effort to be understood. Software engineers would rather review graphics that display important concepts and relationships than comb through lists of metric values.

A single metric can provide insight into a question a developer has about his program or process. For example, a developer might search for the largest module, which could be determined by the Lines of Code (LOC) metric. Tougher questions concerning the testability, reliability or maintainability of software need more information. Metrics can be combined to uncover possible insights into such questions. The next step is to categorize modules by metric patterns, or module signatures, that can present important analysis information to a developer [1].

## 2. MMS Visualization Approach

We begin the visualization process with the primary organizational unit of a software project, a file. Contained within these files are modules, our second organizational level. Depending on the language, a module can be one of many types (procedure, task, generic, function, method, class, etc.). Metrics calculated from a module's particular structure and content will comprise our

third organizational view. Composite metrics require an extension to the metric view. From all of the modules in a project, users must be able to identify which modules to "pay attention to" and the metric combinations that highlighted those modules. This information is encoded into the graphical representation by use of colors, position and table values. Finally, all of this information is time dependent and so time slices provide an additional organizational view in the tool.

MMS Visualization input consists of multiple XML files. DTD files provide the information content for the organizational structure of the software and other general analysis information used during visualization. XML documents containing data conforming to the schema are the unique data for the visualization.

MMS Visualization provides insight into the different dimensions of viewing metrics and software structure as part of the development process through tables and radar charts. The tool can be useful for tasks such as fault localization and system evolution tracing. From the project table, a software engineer can quickly observe the number of files, the number of modules in each file and the module's relative stress (error proneness) indication. At a glance, the distribution of stress (e.g. "red" modules) in the entire project can be viewed. Multiple time slices can be displayed within the project table. Module insertions, deletions and changes are noted in the table, providing an overview of development activities. Individual module metric analyses can be done quickly by selecting a module from the project table. The metrics table or radar view provides the individual metric values and the relative significance among those values.

## 3. Reference

- [1] D. Zage and W. Zage, "A Stress-Point Resolution System Based on Module Signatures", *Proceedings of the 28<sup>th</sup> Annual IEEE/NASA Software Engineering Workshop*, Greenbelt, MD, December 3-4, 2003.