A Framework for Source Code Metrics

Neli Maneva, Software Engineering Department, IMI, BAS
Nikolay Grozev, Musala Soft Ltd
Delyan Lilov, Musala Soft Ltd

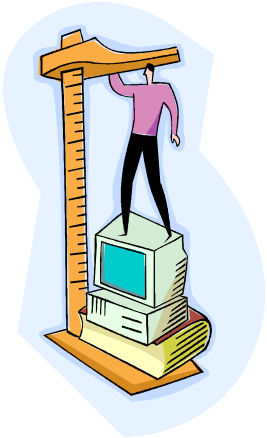
International Conference on Computer Systems and Technologies - CompSysTech'10, 17-18 June 2010

Overview

- Introduction
- Background and Analysis
- Source code framework
- Prototype and validation
- Conclusion and future work
- Authors and Acknowledgements
- Questions

INTRODUCTION

Introduction



- Why use static source code metrics?
  - Source code - an essential part of every software system
  - Static analysis is a useful best practice
  - Solid theoretical background
  - Monitoring and assessment through metrics
- Metrics are rarely used in practice – this is for a reason!
- Our goals:
  - Analyze metrics usage problems
  - Define requirements for a metrics framework
  - Design an abstract framework, meeting the requirements
  - Prototype and validate the framework

BACKGROUND AND ANALYSIS

Background and Analysis

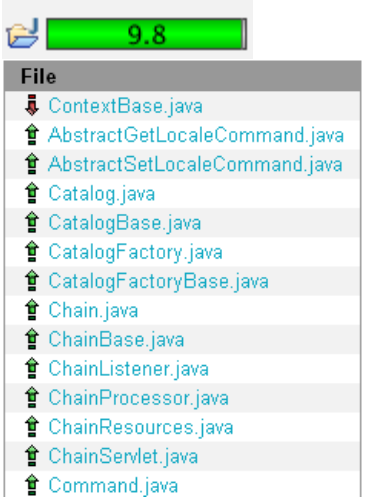
Classification of existing tools

- Reporting tools
  - Compute metrics values and produce reports
  - The user must know details about all metrics
- Combining tools
  - Besides metrics values, produce “combined” evaluations
  - Removes the need to know all about the metrics

Metrics - /net.sourceforge.metrics

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
Number of Packages	16					
Number of Methods (avg/max per type)	1310	6.65	8.553	76	/net.sourceforge.metrics/tgsrc/com/touchgrap...	
tgsrc	489	7.191	11.544	76	/net.sourceforge.metrics/tgsrc/com/touchgrap...	
src	761	6.238	6.553	45	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.core.sources	108	15.429	12.129	45	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.ui	77	9.625	10.111	33	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.core	198	6.6	7.093	27	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.ui.preferences	52	6.5	7.467	26	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.ui.dependencies	95	5.588	3.727	15	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.internal.persistence	18	4.5	4.33	12	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.internal.prevayler.implementa...	54	5.4	2.871	10	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.internal.xml	41	4.1	2.022	9	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.calculators	79	4.158	2.254	8	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.propagators	31	5.167	1.067	7	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.internal.tests	8	2.667	1.886	4	/net.sourceforge.metrics/src/net/sourceforge/...	
net.sourceforge.metrics.internal.prevayler	0	0	0			
classycle	60	8.571	2.556	13	/net.sourceforge.metrics/classycle/classycle/g...	
Lines of Code (avg/max per type)	6593	33.467	49.02	339	/net.sourceforge.metrics/tgsrc/com/touchgrap...	
Number of Interfaces (avg/max per packageFragment)	16	1	1.414	4	/net.sourceforge.metrics/src/net/sourceforge/...	
Lines of Code (avg/max per method)	6593	4.812	7.355	69	/net.sourceforge.metrics/classycle/classycle/g...	calculateAttributes
classycle	324	5.4	9.94	69	/net.sourceforge.metrics/classycle/classycle/g...	calculateAttributes
tgsrc	2321	4.661	8.278	59	/net.sourceforge.metrics/tgsrc/com/touchgrap...	scrollSelectPanel
src	3948	4.862	6.473	52	/net.sourceforge.metrics/src/net/sourceforge/...	setMetrics
net.sourceforge.metrics.ui	544	6.8	8.707	52	/net.sourceforge.metrics/src/net/sourceforge/...	setMetrics
MetricsTable.java	194	10.778	13.831	52	/net.sourceforge.metrics/src/net/sourceforge/...	setMetrics
MetricsTable	194	10.778	13.831	52	/net.sourceforge.metrics/src/net/sourceforge/...	setMetrics
setMetrics	52					

Screenshot from Eclipse Metrics Plugin



Screenshot from Enerjy

Background and Analysis

Problems with existing tools

- Both reporting and combining tools are “hardcoded” in nature
  - Almost no settings available
  - Everything is measured and interpreted in the same way
- Should we really measure everything with the same “scales”?

Background and Analysis

Context and its aspects

- Source code metrics tools should:
  - Extract metrics values with regards to *what is being evaluated*.
  - Combine metrics values with regards to *what is being evaluated*.
- Context – additional info about *what is being evaluated*:
  - Programming languages
  - Used technologies and libraries
  - Project size
  - Architecture
  - Application area
  - Etc.

Conclusion: *A successful source code metrics tool should be "context sensitive"*

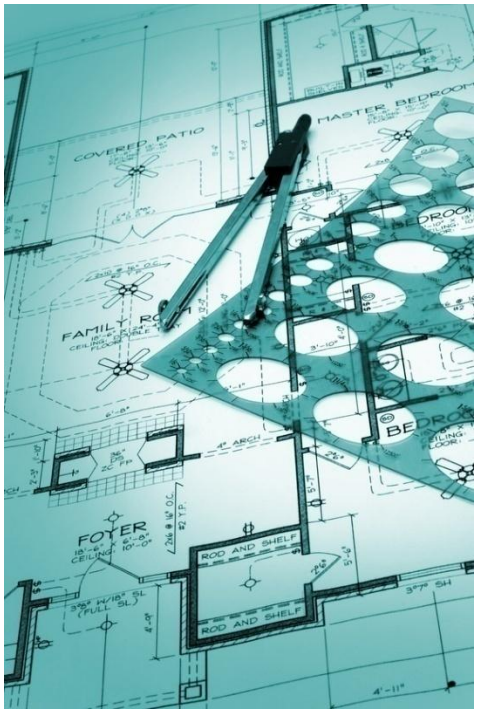
Background and Analysis

Objectives

- Define an abstract/general framework for evaluating source code quality
- It should flexibly accommodate:
  - Contextual extraction of metrics values
  - Contextual combinations of metrics values
- It should provide a solid basis for practical tools through a set of extension points
- Provide some speculations about possible extensions of the ideas of the framework
- Validate the feasibility of the approaches in practice

SOURCE CODE FRAMEWORK

DESIGN AND STRUCTURE



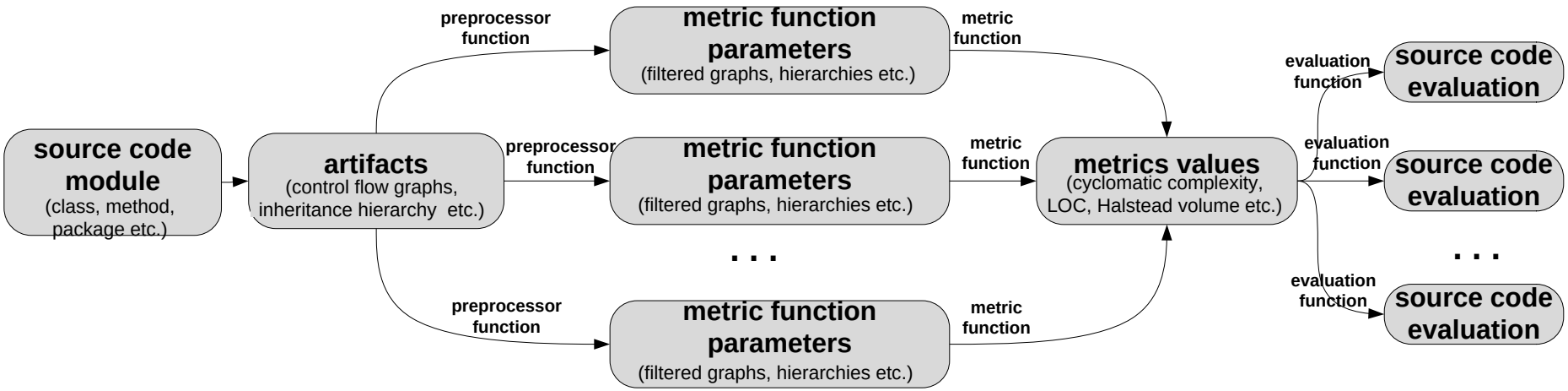
Source code framework

Design and structure

- Base set of metrics
  - Toolbox of metrics
  - Basis for all evaluations
- A stepwise framework (evaluation scheme) that simulates an expert's work of evaluating source code through metrics
- Modeling each step as a function:
  - Metric functions – extract the value of a single metric
  - Preprocessor functions – used to prepare the parameters for the metric functions
  - Evaluation functions – combine the values of metrics into a meaningful source code evaluation

Source code framework

Design and structure

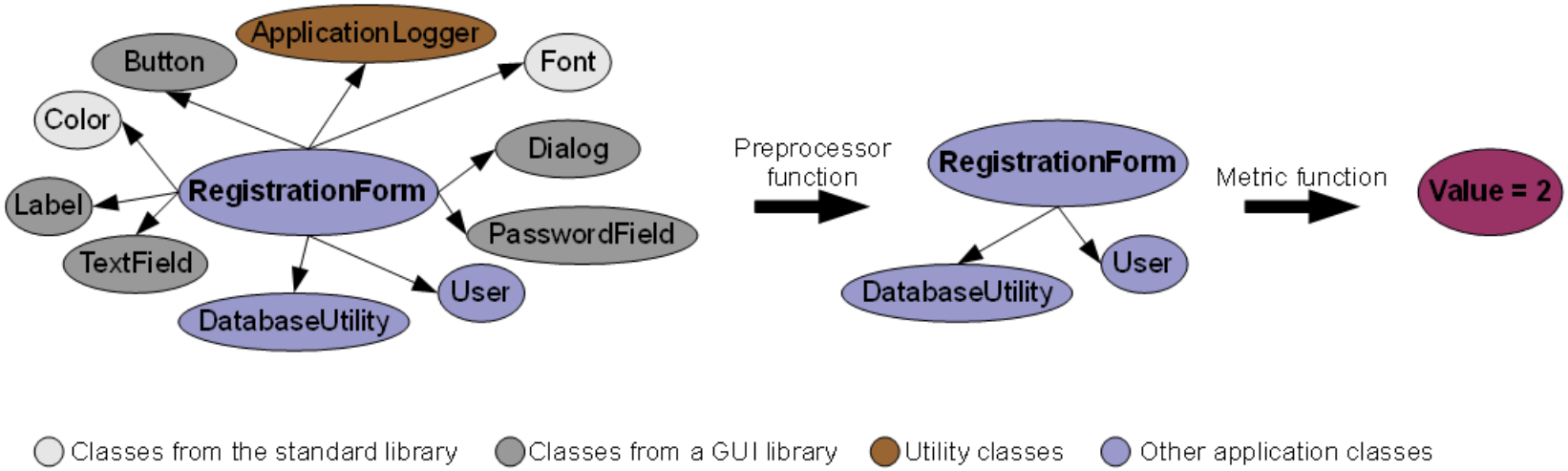


- Contextual user specified logic is “hooked” by:
  - Preprocessor functions – filter the data for the metrics, less “noisy” values
  - Evaluation functions – combine the values of the metrics, benefits from the less “noisy” values

Source code framework

Design and structure – preprocessor functions

- Preprocessor functions:
  - Determine which artifacts are relevant for the computation of a metric
  - Usually such functions filter the irrelevant to a metric input elements
  - Using preprocessor functions results in more “accurate” metric values



Source code framework

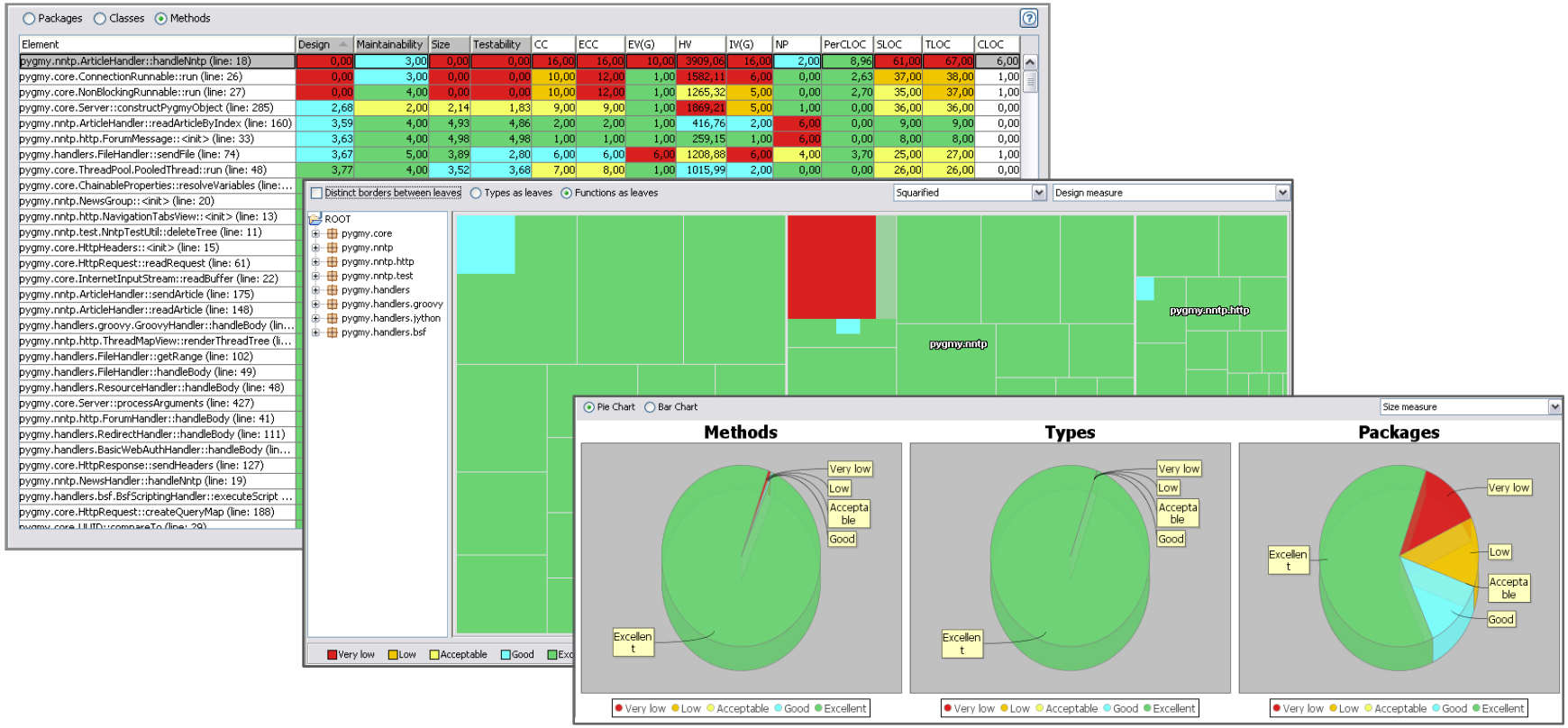
Design and structure – evaluation functions

- Functions highlighting design problems in the code
  - Promising researches into the area of OOP design problem recognition using
  - Can be improved by the less “noise” after the usage of preprocessors
- Functions combining metrics values into a new numerical value
  - Can be modeled as real valued functions
  - Linear combinations vs. Machine learning techniques

PROTOTYPE AND VALIDATION

Prototype

Smart source code analyzer (SSA)



- Eclipse plug-in
- A tool, based on an implementation of the framework
- Also used for validation purposes

Practical validation

- Validation by prototype usage:
  - Used to analyze many open source projects
  - Used in real life development – code quality assessment, code reviews
- Validation results:
  - Users like getting aggregated information and being abstracted from the details of the different metrics
  - Quicker code reviews
  - Problems with setting the contextual information – false positives
  - Problems understanding "*Why this is bad?*"
- Problems can be overcome through additional functionalities, planned for the prototype.

CONCLUSION AND FUTURE WORK

Conclusion

- An analysis of existing tools and approaches was briefed
- A general framework for evaluating source code through metrics was described.
- A prototype was built and used for validation

- Future work:
  - Methods for metrics preprocessing and combinations
  - Visualization techniques
  - Incorporation throughout the software lifecycle

Authors and Acknowledgements

Assoc.Prof. PhD Neli Maneva, Software Engineering Department, Institute of Mathematics and Informatics - BAS neli.maneva@gmail.com

Nikolay Grozev, Musala Soft Ltd, nikolay.grozev@musala.com

Delyan Lilov, Musala Soft Ltd, delyan.lilov@musala.com

This work was partially supported by the National Innovative Fund attached to the Bulgarian Ministry of Economy and Energy (project № 5ИФ-02-3 / 03.12.08).

Questions

Thank you!