Fuzzy System for Software Reliability Classification using Object Oriented Metrics

University of Michigan Dearborn
Presented by Juan C. Garcia
Fall 2010

Outline

- Introduction
- Software Requirements
- Based-Rule Fuzzy System Design
- Fuzzy sets
- Fuzzy Rules
- Experiments and Results
- Conclusion
- Q&A

Introduction

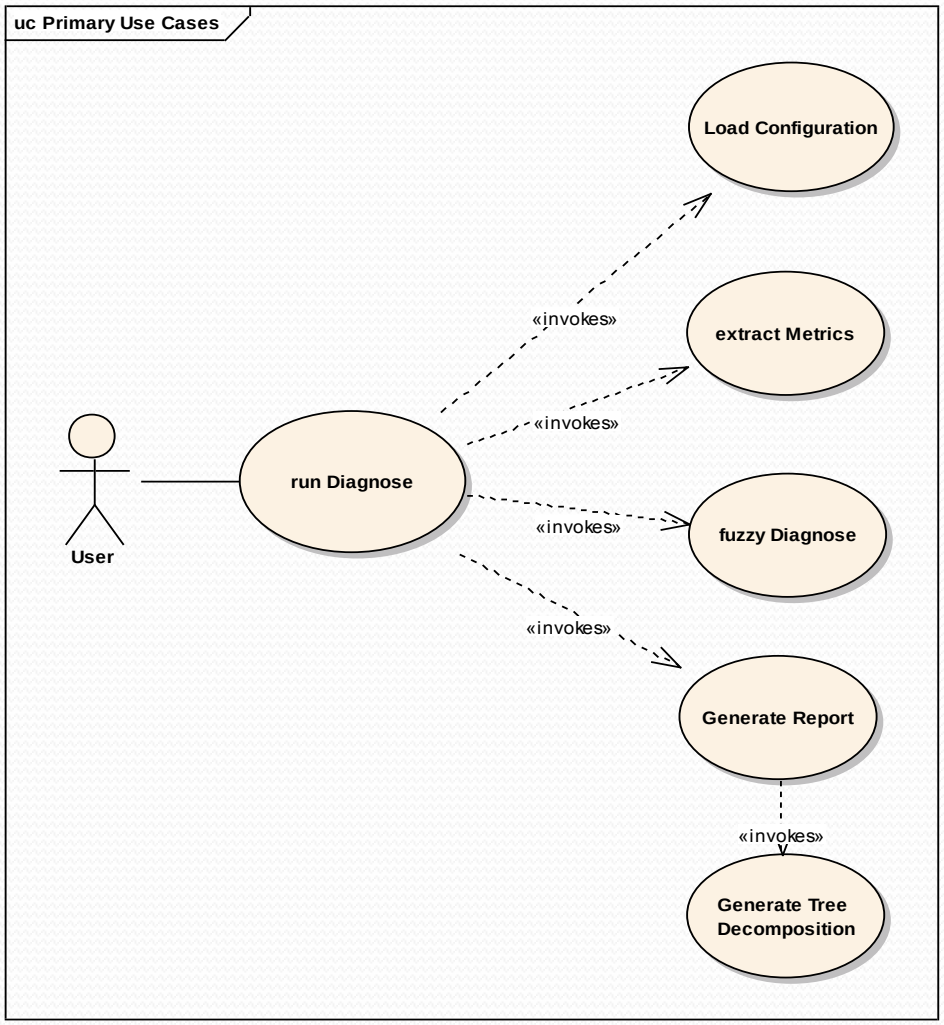
- Managers have been focusing on process improvement in the software development area.
- Metrics focus in OO concepts like classes, objects, instance, method, message passing, inheritance, multiple inheritance, abstraction, encapsulation, polymorphism, decoupling, complexity.
- OO metrics measure computational complexity as well as psychological complexity factors that affect the ability of a programmer to create, comprehend, modify and maintain software.
- They provide a quantitative basis for the development and validation of models during the software development process
- They are used to monitor the software and its areas that need corrective or management action to improve productivity, reliability, usability, testability and quality

Software Requirements

- System to classify java applications based on OO metrics.
  - Automatic extraction of OO metrics from java code
  - Default/base set of rules
  - XML configuration and Framework Application for final users to adjust/add/remove:
    - Fuzzy sets
    - Fuzzy Rules
    - OO Metrics
  - General classification of the software application and fine grain classification of the java classes based on OO Metrics.
  - GUI and command interfaces provided.

Based-Rule Fuzzy System Design

Use cases

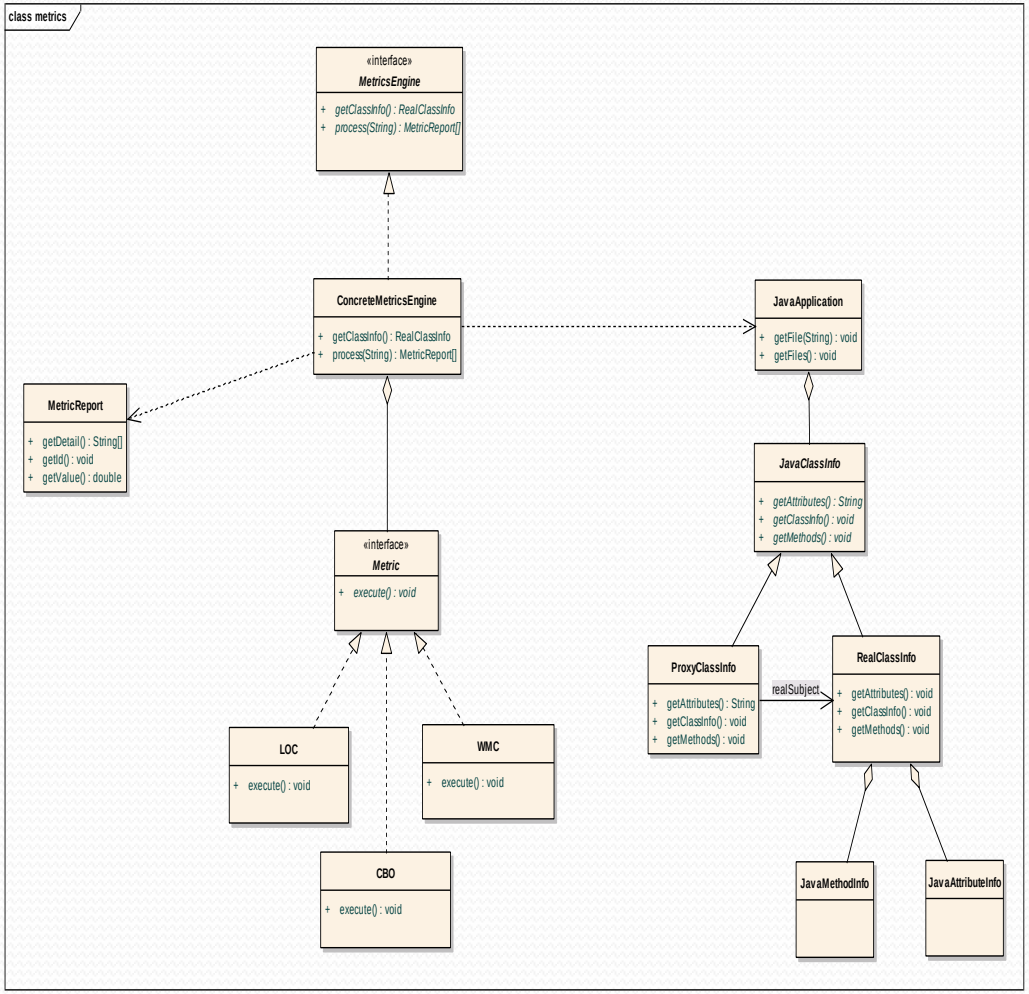


Use Cases

- Load Configuration
- Extract Metrics
- Fuzzy Diagnose
- Generation Report
- Decomposition tree Generation

Based-Rule Fuzzy System Design

OO Metrics Framework



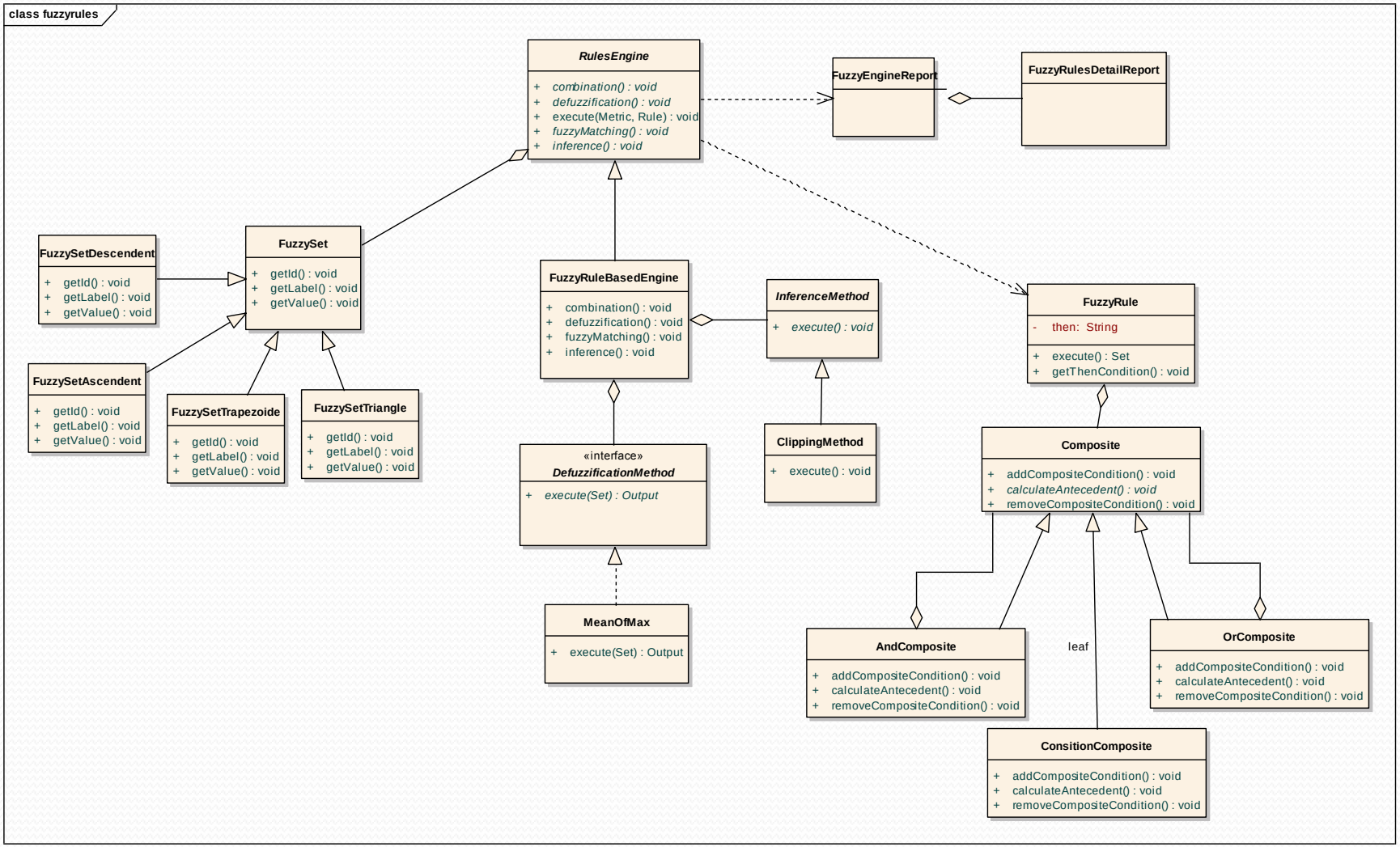
Metrics Engine Framework

OO Metrics can be added or removed depending on the needs of the final user.

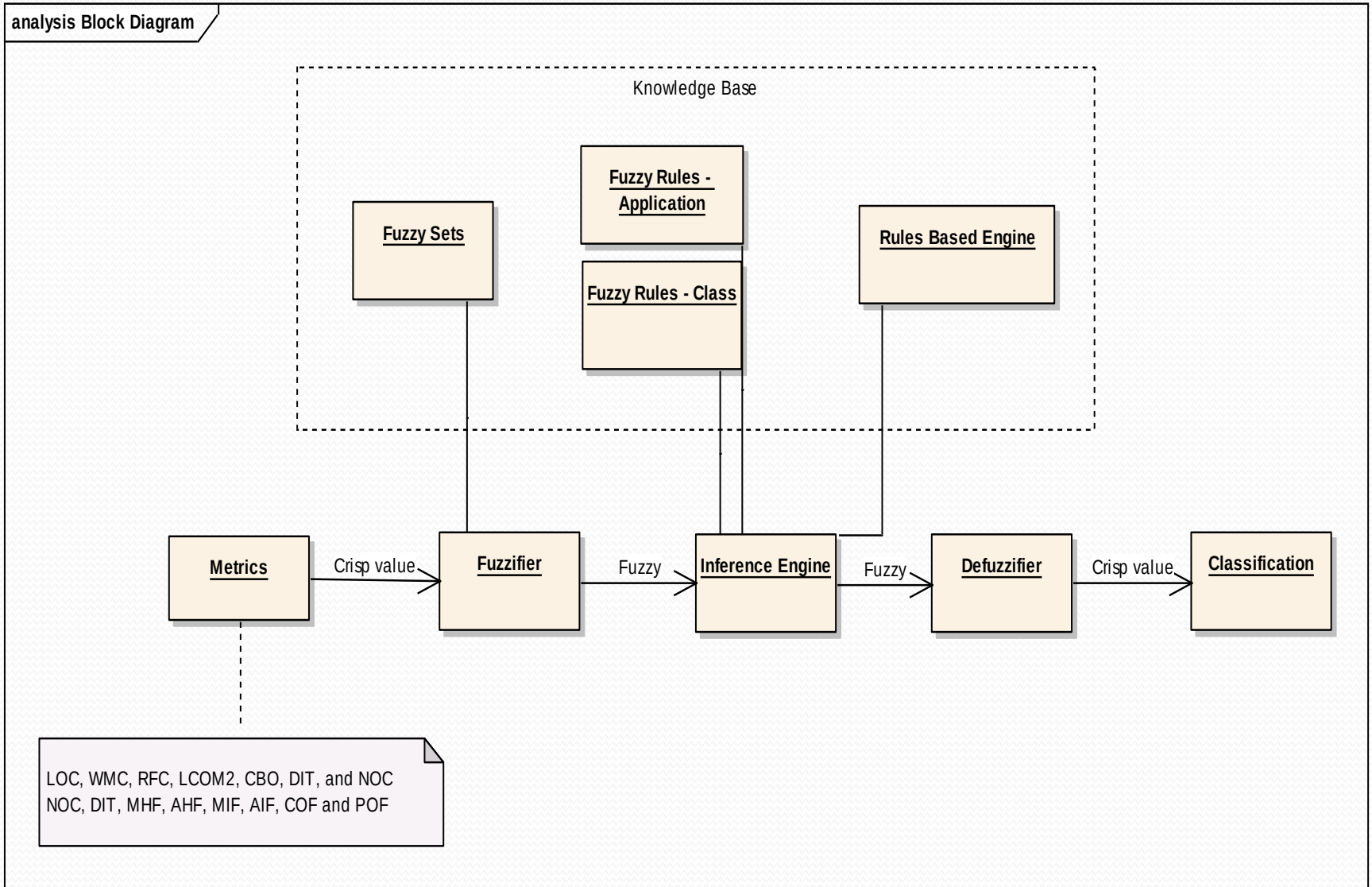
For new OO Metrics the Metric Interface will need to be implemented in the new class and added to the configuration file

Based-Rule Fuzzy System Design

Fuzzy Engine



Fuzzy System Design – Block Diagram



Fuzzy Inputs - OO Metrics

Measurement	Metrics
Complexity	LOC – Lines of Code WMC – Weighted Methods Per Class RFC - Response for a Class
Cohesion	LCOM2 - Lack of Cohesion
Coupling	CBO – Coupling between Object Classes COF – Coupling factor
Inheritance Tree	NOC - Number of Children DIT - Depth of Inheritance Tree
Encapsulation	MHF - Method Hiding Factor AHF - Attribute Hiding Factor
Class inheritance	MIF - Method Inheritance Factor AIF - Attribute Inheritance Factor
Polymorphism	POF - Polymorphism Factor

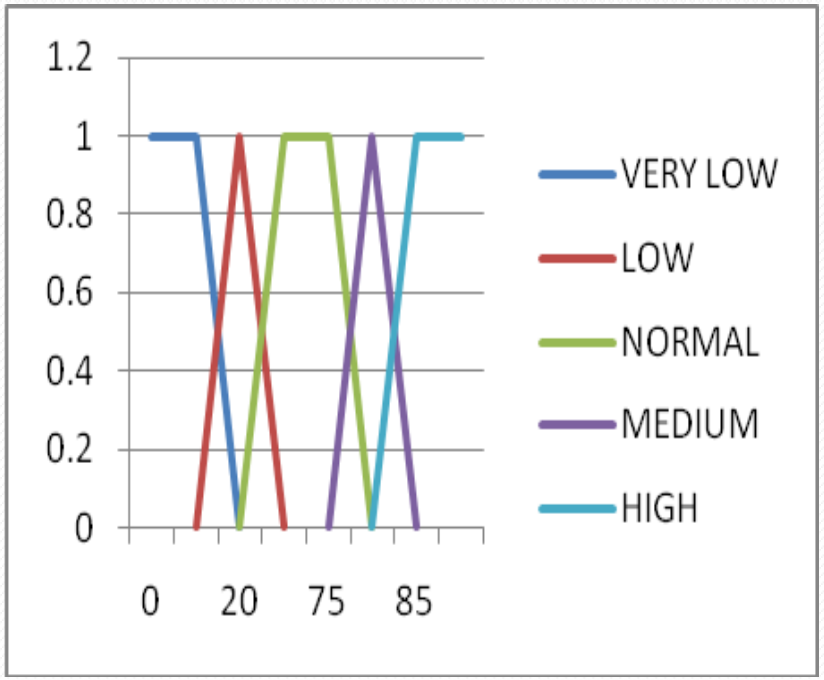
Fuzzy Outputs and Definitions

- Output - OODC= Software classification.
- Definitions:
  - C- Critical
  - H – High
  - M – Medium
  - N - Normal
  - L - Low
  - VL – Very Low

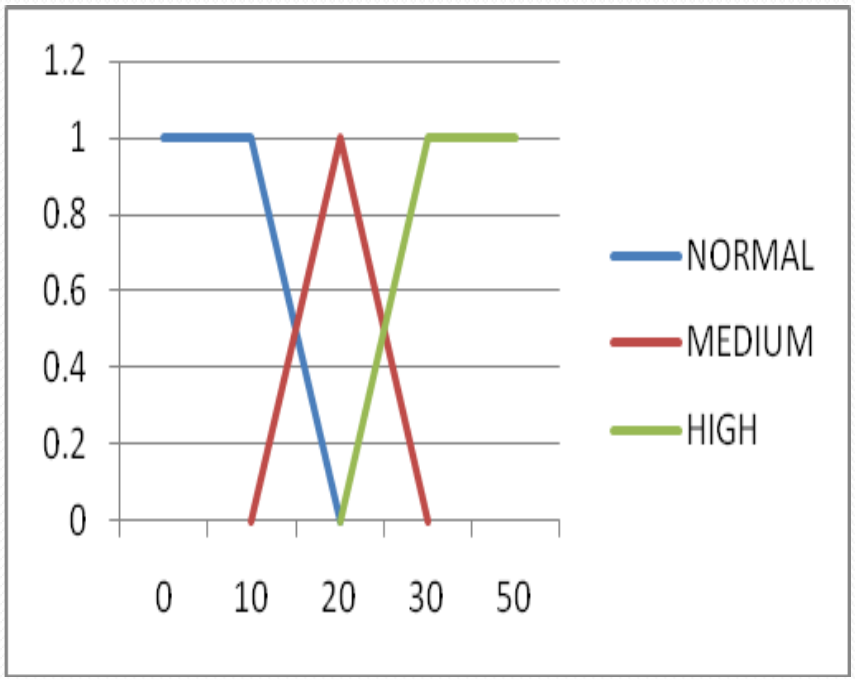
Fuzzy Membership Functions

OO Metrics	Fuzzy Membership Functions
WMC	normal (x:10,20), medium (x:10,20,30) and high(x:20,30)
RFC	normal (x:30,40), medium (x:30,40,50) and high (x:40, 50)
LCOM2	normal (x:60,70), medium (x:60,70,80) and high (x:70,80)
CBO	normal (x: 5, 10), medium(x: 5, 10, 15), high(x: 10, 15)
DIT	normal (x:3,6), medium (x:3,6,9) and high (x:6,9)
NOC	normal (x:10,20), medium (x:10,20,30) and high(20,30)
LOC	(x:750,1000), medium (x:750,1000,1250) and high (x:1000,1250)
MHF	very low (x:5,10), low (x:5,10,15), normal (x:10,15,20,25), medium(20,25,30) and high (x:25,30)
AHF	normal (x:80,90), medium (70,80,90) and high (70,80)
MIF	very low(x:15,20), low(15,20,25), normal(x:20,25,75,80), medium (x:75,80,85) and high (80,85)
AIF	normal(x:40,50), medium(x:40,50,55,65) and high (x:55,65)
COF	normal (x:10,20), medium (x:10,20,30) and high (x:20,30)
POF	rmal (x:0,10), medium (x:0,10,20) and high (x:10,20)

Fuzzy Membership Functions (Cont..) Example



MIF



COF

Fuzzy Rules – Groups Definition

Group #	Metrics	Objective
1	LOC, WMC, RFC	Complexity
2	DIT, NOC	Hierarchical tree.
3	LCOM2	Cohesion
4	CBO, CFO	Coupling
5	MIF, AIF	Inheritance
6	MHF, AHF	Encapsulation
7	POF	Polymorphism

Application Classification:

MHF, AHF, MIF, AIF, COF, POF, DIT, NOC

Single Java Class Classification:

NOC, DIT, LOC, WMC, RFC, LCOM2, CBO

Fuzzy Rules - Conditions (Cont…)

Classification	Fuzzy Conditions
Critical	At least three of the clusters being evaluated have high value;
High	At least two of the clusters being evaluated have high value;
Medium	At least one of the clusters being evaluated have high value;

Fuzzy Rules Single Java Class Classification

Rule#	COMPLEXITY			INHERITANCE TREE		COHESION	COUPLING	Results
	LOC	WMC	RFC	DIT	NOC	LCOM2	CBO	
1	H			H|M|N		H|M|N	H|M|N	medium
2	H|M|N			H		H|M|N	H|M|N	medium
3	H|M|N			H|M|N		H	H|M|N	medium
4	H|M|N			H|M|N		H|M|N	H	medium
5	H			H		H|M|N	H|M|N	high
6	H			H|M|N		H	H|M|N	high
7	H			H|M|N		H|M|N	H	high
8	H|M|N			H		H	H|M|N	high
9	H|M|N			H		H|M|N	H	high
10	H|M|N			H|M|N		H	H	high
11	H			H		H	H|M|N	critical
12	H			H		H|M|N	H	critical
13	H			H|M|N		H	H	critical
14	H|M|N			H		H	H	critical

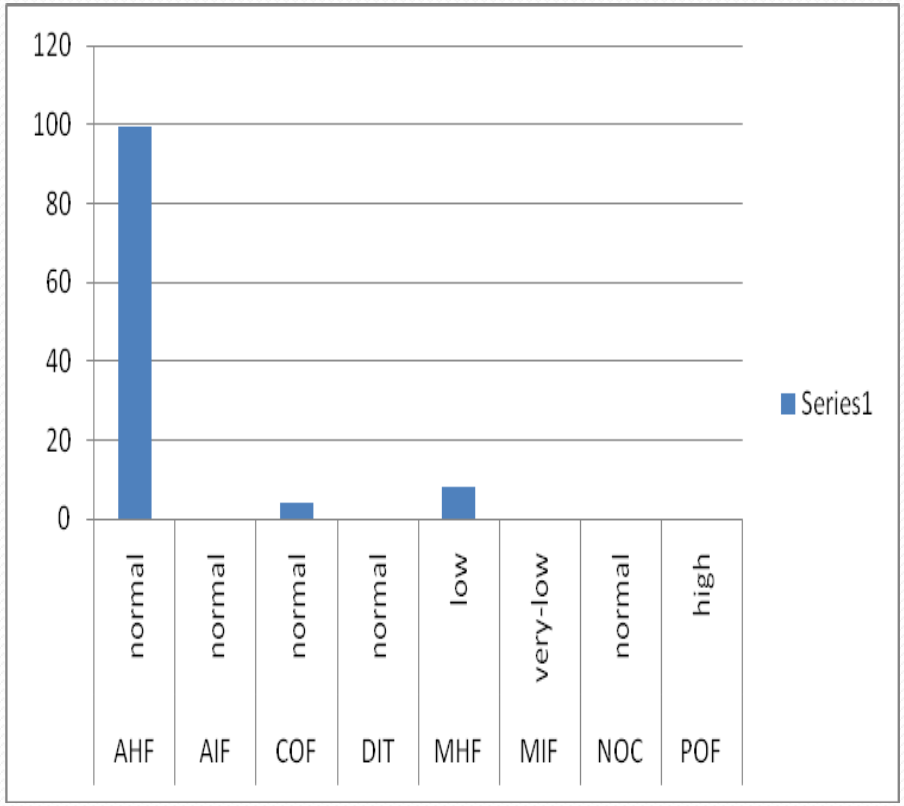
Fuzzy Rules Application Classification

Rule#	Inher. Tree		Inher. Class		Encapsulation		Coupling	Polimorphism	Results
	NOC	DIT	MIF	AIF	MHF	AHF	COF	POF	
1	H		H|M|N		H|M|N		H|M|N	H|M|N	medium
2	H|M|N		H		H|M|N		H|M|N	H|M|N	medium
3	H|M|N		H|M|N		H		H|M|N	H|M|N	medium
4	H|M|N		H|M|N		H|M|N		H	H|M|N	medium
5	H|M|N		H|M|N		H|M|N		H|M|N	H	medium
6	H		H		H|M|N		H|M|N	H|M|N	high
7	H		H|M|N		H		H|M|N	H|M|N	high
8	H		H|M|N		H|M|N		H	H|M|N	high
9	H		H|M|N		H|M|N		H|M|N	H	high
10	H|M|N		H		H		H|M|N	H|M|N	high
11	H|M|N		H		H|M|N		H	H|M|N	high
12	H|M|N		H		H|M|N		H|M|N	H	high
13	H|M|N		H|M|N		H		H	H|M|N	high
14	H|M|N		H|M|N		H		H|M|N	H	high
15	H|M|N		H|M|N		H|M|N		H	H	high
16	H		H		H		H|M|N	H|M|N	critical
17	H		H		H|M|N		H	H|M|N	critical
18	H		H		H|M|N		H|M|N	H	critical
19	H		H|M|N		H		H	H|M|N	critical
20	H		H|M|N		H		H|M|N	H	critical
21	H		H|M|N		H|M|N		H	H	critical
22	H|M|N		H		H		H	H|M|N	critical
23	H|M|N		H		H		H|M|N	H	critical
24	H|M|N		H|M|N		H		H	H	critical

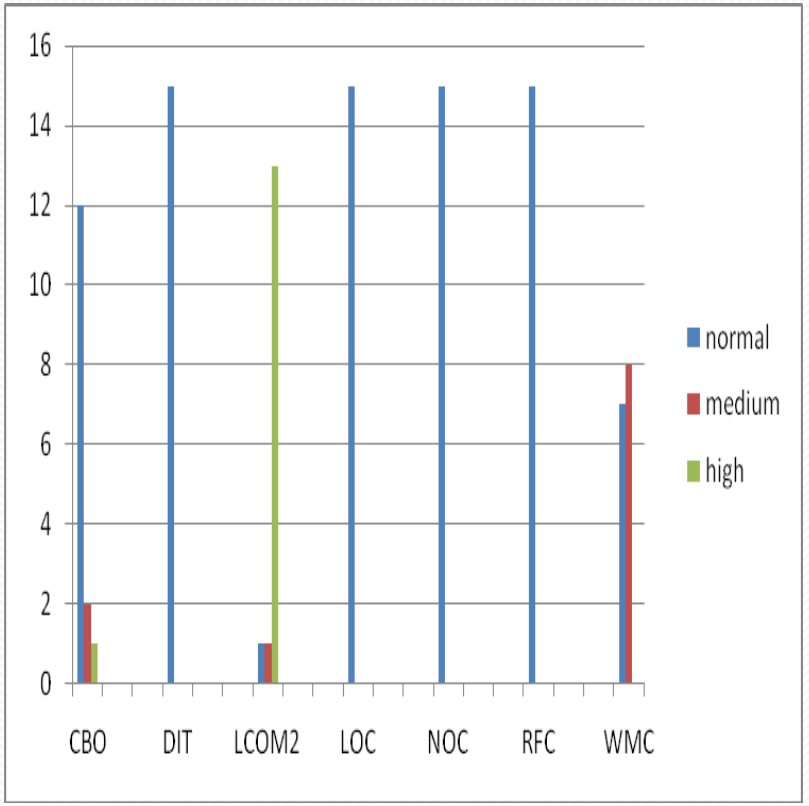
Experiments & results – OO Diagnose Application

- Classification- Normal

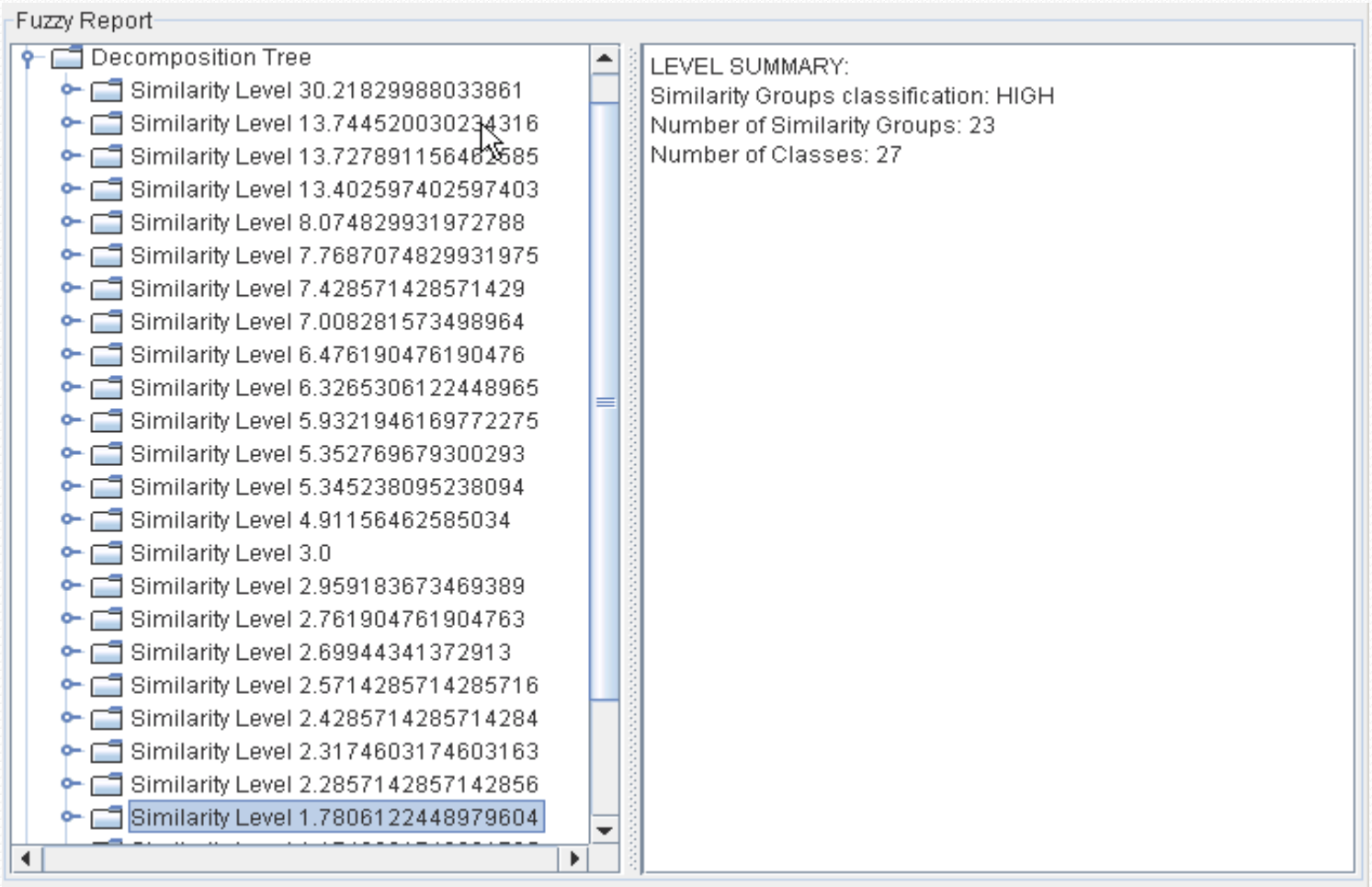
Application



Java Classes



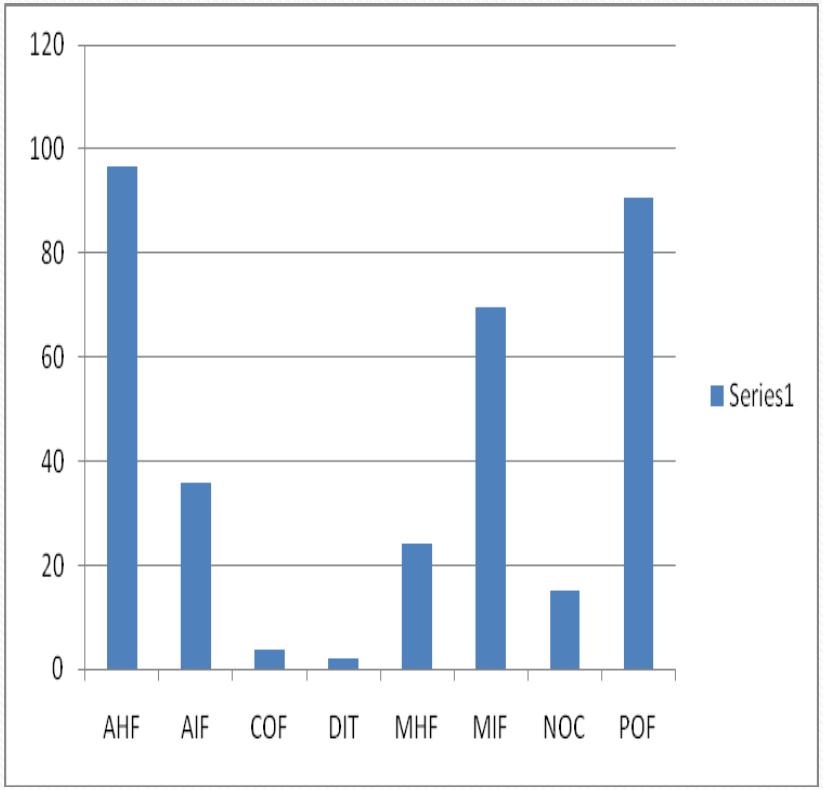
Decomposition Tree – OO Diagnose App



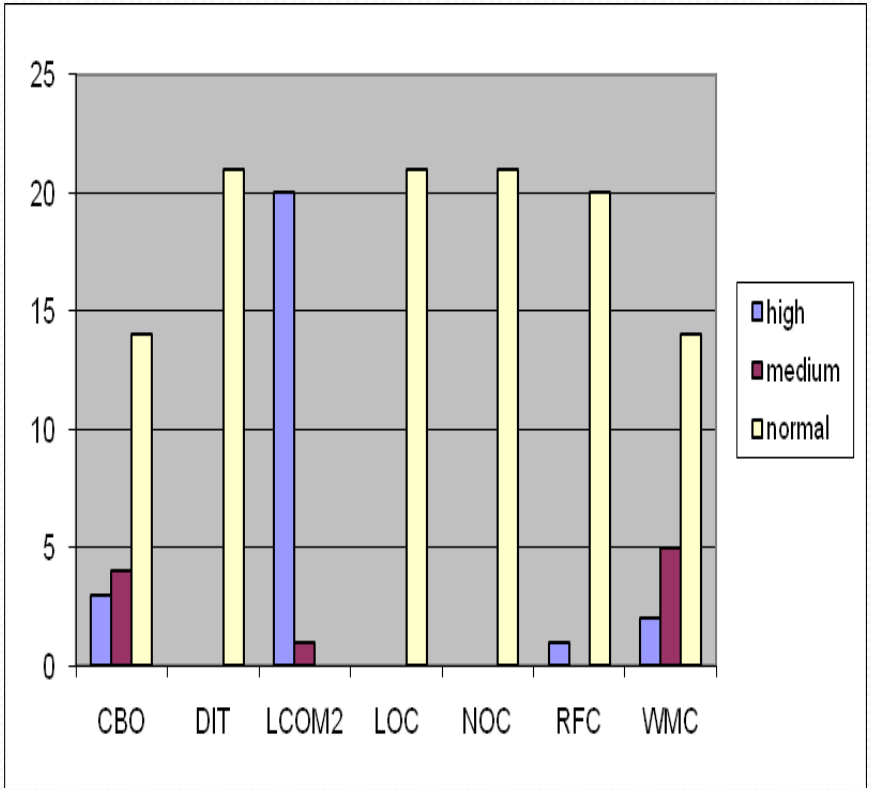
Experiments & Results – ETLF App

- Classification - High

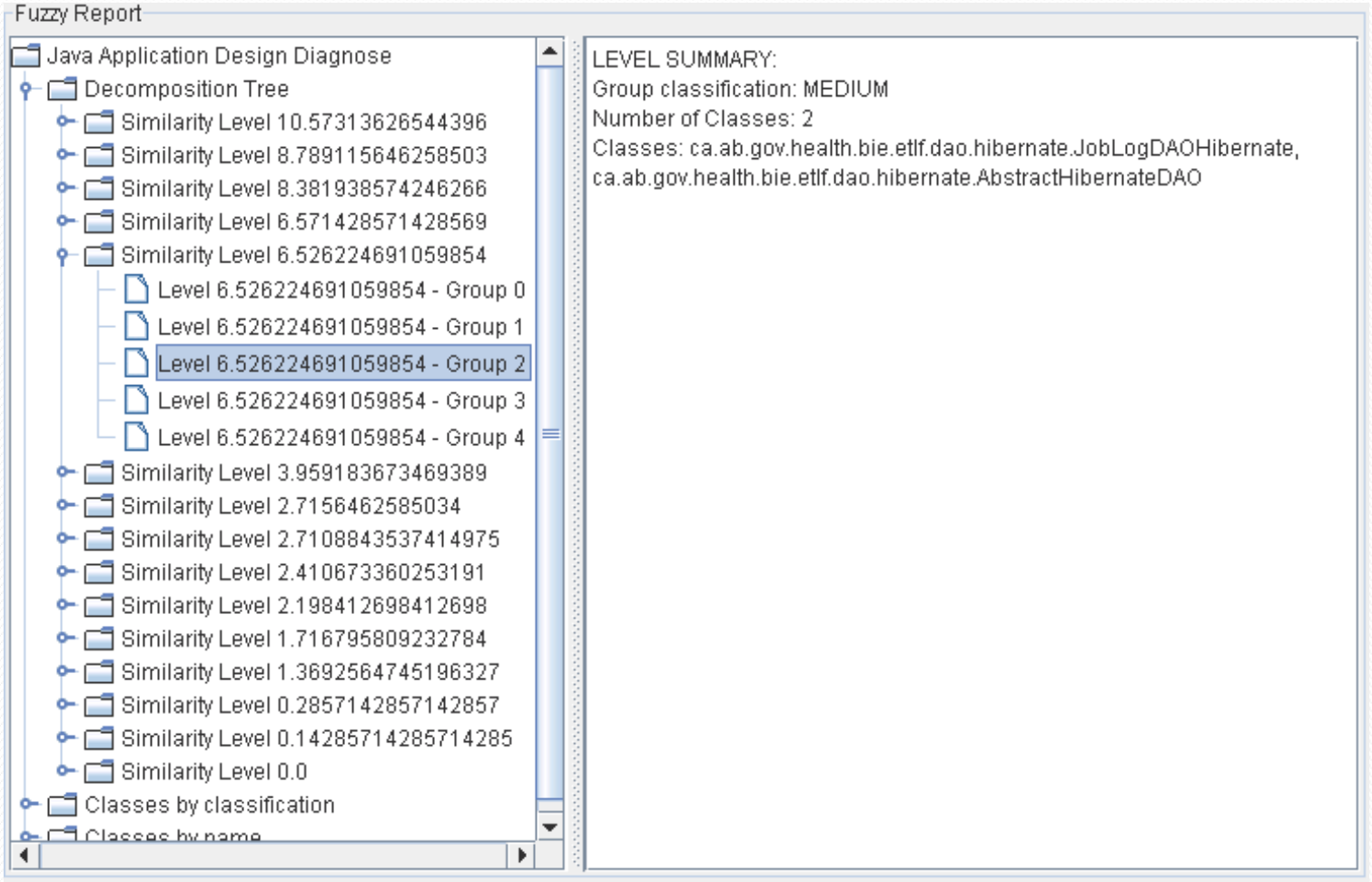
Application



Java Classes



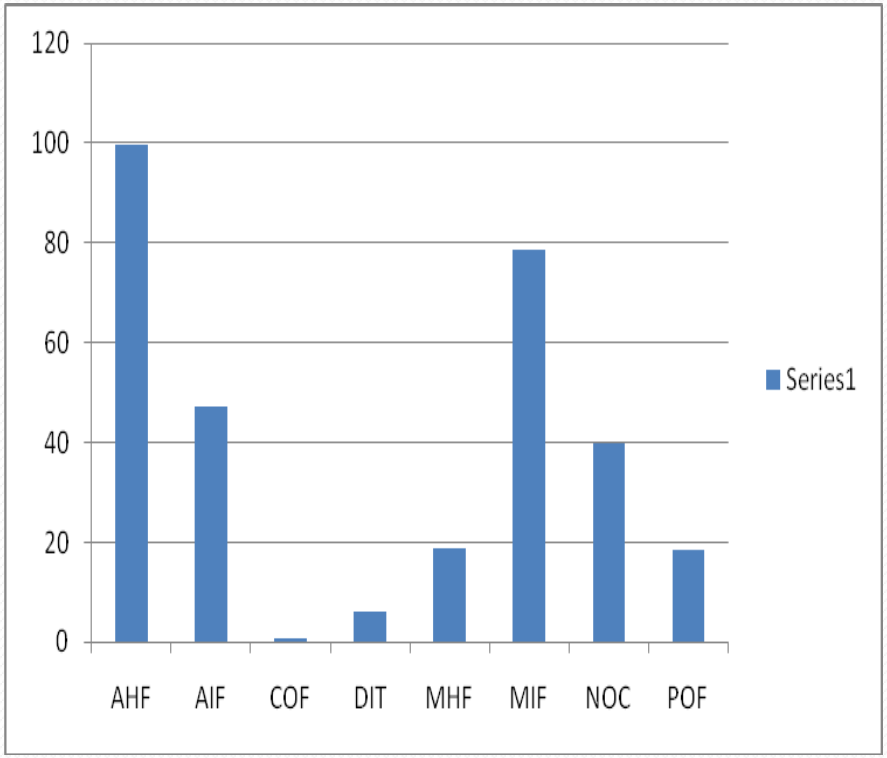
Decomposition Tree – ETLF App



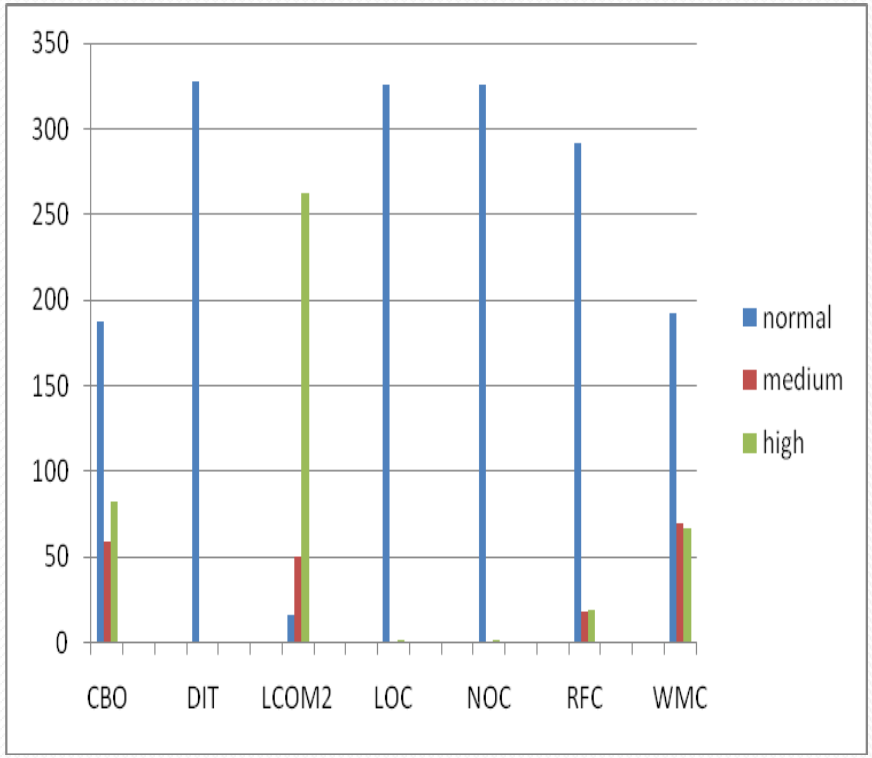
Experiments & Results – BIE Portal

- Classification - Medium

Application



Java Classes



Conclusion

- Appropriate default set of fuzzy sets and fuzzy rules to classify java applications.
- Results are reliable and no additional analysis is needed.
- The decomposition tree useful to address issues with classes with similar metric values.
- The results help enforce the use of object oriented principles in the design and development of java applications.
- Fuzzy system does not prevent metrics to report false-positives and manual analysis is needed for abnormal results.
- The system does not report low values in NOC and DIT at the application level, modifications of these rules and fuzzy sets are expected if needed by the user.
- The fuzzy rules did not entirely utilize medium and normal membership functions, but included if more accurate results are needed.
- System provides objective results because they contain information from statistical sources and several human experts in contrast to human analysis that is bias and can vary depending on the knowledge and experience of the expert.

Q&A