Formal automated software measurement plan.

Sarah DAHAB
Telecom SudParis, CNRS UMR 5157, Univ. Paris-Saclay, France

Supervisor: Prof. Stephane MAAG
Plan

- Context
- Problematics
- Motivations & Objectives
- State of Art
- Our approach
- Conclusion & Perspectives
Context

- **Software measurement**
  - More complex systems
  - Current measurement process is no longer adapted:
    - Sequential
    - Static
    - Lots of data to manage and analyze
  - Heavy measurement process
  - High demand for adapted measurement

→ Improvement of software measurement process
Problematics

- **Lack of formal basis in SW measurement**
  - Formal metric definition & Measurement plan

- **Heavier measurement management load**
  - Static measurements plan
    - Planed at the beginning
  - Sequential analysis
  - Resources and Time consumption/cost
  - Useless metrics (at some times)

- **Huge amount of data to analyze**
  - More properties to be evaluated (system complexity)
  - Need of great measurement coverage
  - Difficulties to find the properties of interest (cross information...)

- **Expert-dependent**
Motivations & Objectives

Increase the measurement process quality, efficiency and reliability

- Improve the metrics sustainability, use and interoperability
- Reduce the software measurement management costs
- Ensure an accurate evaluation continuously

- **Standard-driven Metrics**
  - Standard specification
  - SMM of OMG

- **Automated analysis**
  - Machine learning – measurements result classifications
  - @Runtime

- **Metrics suggestion**
  - Measurement plan flexible
  - Accurate measurement @Runtime
  - Wide coverage continuously
Our objectives

- Metric 1
- Metric 2
- Metric 3

Measurement Plan
- Measure X
- Measure Y
- Measure Z

Software Measurand

Measures
1. Measures 1
2. Measures 2
3. Measures 3

Measurements

Analyses
SVM supervised, ML @runtime

Measurement plan suggested
State of Art

Software evaluation model
- ISO/IEC 25000: 2011 as software quality models
  ➔ the implementation and management is left to the charge of the project manager.
  ➔ Lack of measurement formal models

Software measurement knowledge
  ➔ Difficult to measure the cost of a measurement plan

Machine Learning approach for SW defect prediction
  ➔ Not used for the suggestion

Prioritize security inspection and testing efforts
  ➔ Only use for security and static selection

... And many others works on software measurement
Measurement Definition

- **Measurable space [1]**
  - Element to be measured $X$
  - The set of properties measurable $A$
  - $(X, A) | A \in X$

- **Measure space**
  - Measurable space
  - Associated function
  - $(X, A, f) | A \in X, f : A \rightarrow Z$

- **Software metric**
  - Software property
  - Software scope
  - Software measure

Formal-driven Metrics Specification through Modeling

- Structured Metric Meta-model (SMM) [1]
  - OMG standard specification
  - Meta-model to formally specify
    - software measurement
    - Software metric
  - Standard interchange format, measurement documentation

- Modelio [2]
  - Open source modeling tool
  - Based on UML

The formal Weighted Class Complexity Metric modeled in SMM with Modelio [1]

\[ W_{CC} = Na + \sum_{p=1}^{s} MCp \]

- \( Na \): number of attribute
- \( MCp \): method complexity of CC

\[ W_{CC} = Na + CC \]

Our approach : A Learning Metrics Suggester

- **Our Set of Measures & properties associated**
  - Relevant indicators on software properties state
  - ISO/IEC standard 25000

- **Learning-based analysis**
  - Based on Machine Learning technique
  - Classification of measures result vectors
  - Mandatory Features Selection

- **Analysis-based measurement plan suggestion**
  - Metrics suggestion according to the analysis
Our Set of Measures as Software Properties Indicators

- Maintainability Index
- Cognitive Complexity
- Code Size
- Number of Bugs
- Time Test Running
- Response Time
- Computational Cost
- Communication Cost
- Infrastructure Cost
- Number of Tasks
- Usability
- Illegal Operations
- Stability Response Time
- Precision

- Maintainability
- Performance
- Executive System
- Functionality
Learning-based analysis

- **Support Vector Machine (SVM) [1]**
  - Supervised learning technique
  - Classify a data sample
  - Through a linear hyperplane
  - From a training data set
  ➔ Automating & Supports big data

➔ Effective and expert-independent measurements analysis

Automated analysis: classification

Each line corresponds of a set of different measures result. Each line (set) is classify in one property by the SVM algorithm.
Dynamic Mandatory Features Selection

- Subset of features with good predictive power
  - RFE: Recursive Feature Elimination [1]
  - Based on the classification
    ➞ Dynamic mandatory metrics selection

➳ Keep the minimal overall information

Measurement Plan Suggestion

- **Measurements analysis**
  - Our Set of Measures classification related to its values
  - Using of RFE algorithm to select mandatory features
    ➔ Property of interest & Mandatory features

- **Efficient measures suggestion**
  - Set of metrics associated to the property of interest
    ➔ Specific measurement suggestion
  - Selected mandatory features
    ➔ keeping information on others property

  ➔ **Flexible measurement plan**
Metrics Suggester Framework

Measurement Plan

- Metric 1
- Metric 2
- Metric 3
- Metric 4
- ... (ellipses)
- Metric N

Execution & Collection

The others

Most number contained

Characteristics:

- Characteristic 1
- Characteristic 2
- Characteristic 3
- Characteristic 4

Learning-aided Analysis

The others

Characteristics:

- Characteristic 1
- Characteristic 2
- Characteristic 3
- Characteristic 4

Metrics associated

Selected Metrics Activator

Selected Metrics Inhibitor

Metrics Suggester Procedures

RFE
Metrics Suggester Framework

Measurement Plan
Flexible

Execution & Collection

Metric 1

Metric 2

Metric 3

Metric 4

...  

Metric N

Learning-aided Analysis

Most number contained

The others

Metrics associated

Others

Characteristic 1

Characteristic 2

Characteristics

Metrics for flexible measurement plan

Metrics Suggester Procedures

RFE

Selected Metrics Activator

Selected Metrics Inhibitor
Experiments:

- **Case study**
  - European project MEASURE
  - OO platform in use
  - 15 metrics and 4 properties as measurement basis

- **16,000,000 vectors / 32 subsets of 500,000 vectors**
  - As measurements generation
  - 4 classes
  - 15 features

- **Scenarios**
  - 5 metrics as initial MP defined by the expert
    - Maintainability Index
    - Response Time
    - Running Time
    - Usability
    - Computational Cost
  - Class of interest: the one with the most predicted instances
Each column represents the number of metrics suggested by the classification result.

Each column represents the classification result of the suggested metrics by the last classification. The classification is done between the four classes distinguished by colors.
Metrics Suggester Visualization

[Image of a dashboard with predictions and latest predictions for different classes]

Classifier: SVC(C=2.0, cache_size=200.0, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma=0.125, kernel='rbf', max_iter=1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)

Conclusion & Perspectives

Conclusion:

➢ Expert-independent analysis
  ➢ Automated & Big data analysis at runtime
➢ Flexible measurement process at runtime
➢ Safe measurement coverage continuously

➢ Open questions: Relevant/efficient suggestions? Time and cost reduced? Usefulness to industrial experts (compared to fixed MP)?

Perspectives:

➢ Industrial integration
➢ Measurement gathering intervals modified at runtime
  ➢ Variable measurement cycles
➢ Exploration of other ML techniques
➢ Relevant justification
Bibliography

- Requirements and Evaluation (SQuaRE) -- System and software quality models, March, 2011


Thank you for your attention