

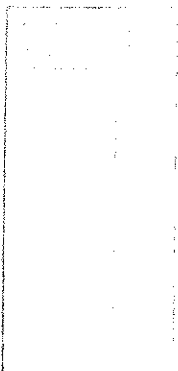
Modeling and Analysis of Metrics Databases

by

Raymond A. Paul

A DISSERTATION PRESENTED TO
THE FACULTY OF UNIVERSITY OF TSUKUBA
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF ENGINEERING

January 1999



*This dissertation is dedicated to the memory
of my mother, Bertha Paul (1917-1986)*

ABSTRACT

In this thesis we investigate important research issues related to quantitative project management in software engineering and propose various approaches to tackle these challenges. Metric databases are among the most useful resources available to project managers to aid in their decision making. We identify a core set of software metrics to be collected for a wide range of projects. These metrics are broadly classified as management, reliability, and quality metrics. Proper analysis of these software metrics and correct interpretation of the analysis results can provide comprehensive information to aid in management decision-making, with the consequent reduction of risks associated with software development projects. For this purpose we propose a framework for integrating various statistical and analytical techniques with software metrics data. The proposed framework can provide a powerful environment for risk monitoring and assessing the impacts of corrective actions and decisions made by the management team.

One of the key contributions of this research is development of two models that can be used to describe the productivity of a software team as a function of effort. A comprehensive analysis of these models based on real data provides a strong validation of these models. The collection of metrics data for past projects and their integration into a metrics database enables us to accumulate past experiences on previous projects and reuse them for analysis of current and planned projects. Management can then pose queries on the metrics database using the appropriate database query language or request

for further analysis of raw metrics. The underlying data model for the DBMS has a direct impact on the ease and efficiency with which such queries are posed, and we therefore examine the effectiveness of different data models with respect to various classes of database queries. With respect to the DBMS data model, we compare various data models and conclude that while the relational model proved adequate for many simple and straightforward queries, alternative models such as object-oriented and graph data models perform better when more complex queries are posed; particularly with queries involving temporal dimension and recursion. Moreover, the metrics database may undergo structural reorganizations for efficiency or other reasons as metrics data for new projects are added to the database and as different queries are identified. The relational model is relatively inflexible to structural modification as it requires the queries to be rephrased; the object-oriented and graph data model, however, does not suffer from such restrictions. The choice of the underlying data model for a metrics database therefore depends very much on the queries the management expects to pose, as well as the type of analysis to be performed.

In order to support broad range of semantic queries we propose a formal framework based on a recursive graph formalism that uses a Petri Net based model. The proposed framework can allow management of software data schema evolution as well as can assist user to construct arbitrary object-oriented views of risk/quality associated with a software project.

ACKNOWLEDGMENTS

I would like to express my heartfelt thanks to my adviser Professor Nobuo Ohbo for providing encouragement, numerous suggestions and guidance on this research. I am thankful to all the Professors on my dissertation committee, for providing me insight into several research ideas. I am also grateful to Professor T. Kunii for his comments on the relational and graph data models and providing me a clear understanding of the importance of the underlying data model with respect to metrics databases

My interest in metrics databases arise from my professional responsibilities at the US Department of Defense, and my numerous interactions concerning metrics with software engineers and program managers of major multinational engineering corporations. Thanks, too, go to my colleagues for their insight as to the importance of metrics databases in project management, and their contributions on new queries and analysis techniques for metrics databases. My interaction with them in these areas has been most helpful in my research.

Professor C. V. Ramamoorthy at the University of California, Berkeley, Professor Y. Shinagawa, Prof. A. Goel at Syracuse University, and Professor A. Ghafoor at Purdue University, have also been most helpful in providing me with additional insight as to related work on software project management and metrics databases, as well as performance comparison techniques for various DBMS data models. I would like to take this opportunity to thank them for their valuable time and effort.

Last but not least, I am most grateful to my wife Alice and my daughter Ann for the concern, understanding, and moral support they have provided me throughout my work.

TABLE OF CONTENTS

Contents	Page
Chapter 1: Introduction	
1.1 Research Objectives	1-1
1.2 The Concept of Software Risk	1-3
1.3 Motivation for Metrics	1-4
1.4 Motivation for Metrics Database	1-8
1.5 Research Issues	1-9
1.5.1 Software Metrics	1-10
1.5.2 Software Metrics Analysis Techniques	1-11
1.5.3 Data Models for Software Metrics	1-13
1.6 Thesis Organization	1-13
Chapter 2: Software Metrics in Project Management	
2.1 Historical Overview	2-2
2.1.1 Cost and Effort Estimation Models	2-3
2.1.2 Metrics Based On System Evolution	2-3
2.1.3 Software Science (Product) Metrics	2-5
2.2 Test and Evaluation Metrics Set	2-6
2.2.1 Objectives	2-6
2.2.2 Selection Criteria	2-8
2.2.3 Customer Satisfaction Measure	2-9
2.2.3.1 Meeting Customer's Requirements	2-10
2.2.4 Classification	2-12
2.2.4.1 Management Metrics	2-12
2.2.4.2 Requirements Metrics	2-12
2.2.4.3 Quality Metrics	2-13
2.2.5 Formal Notation of Metrics Domains	2-15
2.3 Related Work	2-16
2.4 Management Metrics	2-18
2.4.1 Cost Metric	2-19
2.4.2 Schedule Metric	2-21
2.4.3 Computer Resource Utilization Metric	2-23
2.4.4 Software Engineering Environment Metric	2-26
2.5 Requirement Metrics	2-27
2.5.1 Requirement Traceability Metrics	2-27
2.5.2 Requirement Stability Metric	2-29
2.6 Quality Metrics	2-30
2.6.1 Design Stability Metric	2-31
2.6.2 Complexity Metric	2-32
2.6.3 Breadth of Testing Metric	2-36
2.6.4 Depth of Testing Metric	2-37

2.6.5 Fault Profiles Metric	2-37
2.6.6 Reliability Metric	2-39
2.7 Relationships Among Metrics	2-40
2.8 Conceptual Modeling of Software Metrics	2-43
2.9 Sample Data for Software Metrics	2-48
Chapter 3: Queries on Metrics Database	
3.1 Introduction	3-1
3.2 Simple Examination and Classification of Queries for Software Metrics	3-3
3.3 Goal-Question-Metric (GQM)	3-4
3.4 Schedule Scenarios	3-8
3.5 Technical Scenarios	3-12
3.5.1 Modules Requiring Extensive Labor Effort	3-12
3.5.2 Completeness of Testing	3-13
3.6 Cost Scenarios	3-13
3.6.1 Are Costs Under Control?	3-14
3.6.2 Where Are The Resources Going?	3-14
3.7 Acquisition Scenarios	3-16
3.7.1 Technical Maturity	3-16
3.7.2 Knowledge of Requirements	3-17
3.7.3 Adherence to Software Development Plan	3-18
3.7.4 Addressing Technical Complexities	3-19
3.8 Conclusions	3-20
Chapter 4: Analytical Techniques for Metrics Guided Risk Management	
4.1 Introduction	4-1
4.2 Data Dependency Techniques for Software Metrics Data	4-4
4.2.1 E-R Modeling Technique and an E-R Model for T&E Metrics	4-4
4.2.2 Influence Diagrams	4-7
4.2.2.1 Characteristics of Influence Diagrams	4-8
4.2.2.2 Influence Diagrams for Software Metrics Data	4-9
4.3 Analytical Techniques for Software Metrics Data	4-12
4.3.1 Criteria for Selecting Metrics Software Data Analysis Techniques	4-13
4.3.2 Metrics Data Analysis Techniques	4-14
4.3.2.1 Univariate Techniques	4-14
4.3.2.2 Bivariate Techniques	4-16
4.3.2.3 Multivariate Techniques	4-18
4.3.2.4 Multiresolution Analysis	4-20
4.3.3 Summarizing the Overall Process of Metrics Data Analysis	4-25

Chapter 5: Metrics Classification Techniques and an Framework for Risk Management

5.1 Introduction	5-1
5.2 Classification Trees	5-3
5.2.1 What Is A Classification Tree?	5-3
5.2.2 Constructing A Classification Tree	5-5
5.2.2.1 Choosing the Training Set	5-5
5.2.2.2 Partitioning into Ranges	5-5
5.2.2.3 Choosing a Metric at a Node	5-6
5.2.2.4 Partial Tree Construction	5-6
5.2.2.5 Recursive Metric Assignment	5-7
5.2.3 An Example of Classification Tree	5-8
5.2.4 Advantages and Drawbacks of Classification Trees	5-11
5.3 Neural Networks	5-11
5.3.1 Input Phase	5-14
5.3.2 Learning Phase	5-15
5.3.3 Output Phase	5-16
5.3.4 Advantages of Neural Network-Based Classification Methodology	5-16
5.4 An Integrated Software Project Management Framework	5-18
5.5 Conclusion	5-22

Chapter 6: Decision Making Framework for Project Management Using a Value Based Software Engineering Approach

6.1 Introduction	6-1
6.2 Background	6-3
6.3 The Putnam/Norden Model	6-6
6.4 The Proposed Models for Metric-Based Decision/Quality Tradeoffs	6-10
6.4.1 The Rapid Learning Model	6-11
6.4.2 The Effect of Rapid Productivity on the Quality	6-17
6.4.3 Average Productivity Model	6-21
6.4.4 The Effect of Average Productivity on the Quality	6-26
6.5 Discussion	6-29
6.6 Experimental Validation of the Proposed Models	6-33
6.7 Conclusion	6-38

Chapter 7: Data Models for Metrics-Based Project Management Systems

7.1 Introduction	7-1
7.2 Overview of Data Models for Software Metrics Database Management	7-2
7.2.1 Metrics Queries Using the Relational Data Model	7-3
7.2.1.1 Simple Selection Queries	7-3
7.2.1.2 Temporal Queries	7-3
7.2.1.3 Recursive Queries	7-5
7.2.1.4 Strength of the Relational Model for Metrics Databases	7-6

7.2.1.5 Weakness of the Relational Model for Metrics Databases	7-6
7.2.1.6 Extensions to the Relational Model	7-8
7.3 Metrics Queries Using the Object-Oriented Data Model	7-9
7.3.1 Formulation of Metric Database Queries	7-10
7.3.2 Benefits of O-O Model for Metric Queries	7-14
7.3.3 Drawbacks of O-O Model for Metric Queries	7-14
7.4 Software Metrics Queries Using the Graph Data Model	7-16
7.4.1 Sample Software Metrics Queries in GDL	7-17
7.4.2 Benefits and Drawbacks of Graph Data Model for Metric Queries	7-18
7.5 A Formal Approach for Semantic Modeling of Metrics Data for Quality and Risk Management	7-19
7.5.1 A Two-Level Spatio-Temporal Modeling Approach	7-20
7.5.1.1 Modeling in Spatial Domain - Low Level	7-21
7.5.1.2 Modeling in Temporal Domain - High Level	7-22
7.5.1.3 Identification of Scenarios from Metrics Data	7-24
7.5.2 A Petri-Net Based Spatio-Temporal Modeling of Software Metrics	7-28
7.5.2.1 Temporal Semantic Modeling	7-29
7.5.2.2 Petri-Nets	7-29
7.5.2.3 The Proposed Petri-Net Model for Semantic Modeling	7-31
7.5.2.4 Query Evaluation and Execution of SMPN	7-32
7.5.2.5 Examples of SMPN	7-32
7.5.2.6 Representing Standard Database Query Operations Using SMPN	7-35
7.5.2.7 Basic Temporal Relations and the SMPN	7-36
7.5.2.8 Properties of SMPN	7-37
7.5.2.8.1 Precedence Relationships and Partial Ordering	7-38
7.5.2.8.2 Reachability	7-38
7.5.2.8.3 Liveness	7-38
7.5.2.8.4 Boundedness	7-39
7.5.2.8.5 Conservation	7-39
7.5.3 Object-Oriented Modeling of Software Quality/Risk	7-39
7.5.3.1 From Scenario to Objects	7-40
7.6 Performance Considerations for Implementation of Software Database Management System	7-44
7.7 Conclusion	7-46

Chapter 8: Schema Evolution and View Management in Metrics Databases Using Recursive Graphs

8.1 Introduction	8-1
8.2 Schema Evolution in Metrics Databases	8-2

8.2.1 Attribute Addition and Deletion	8-2
8.2.2 Creating New Relationships	8-4
8.2.3 Creating New Entities	8-6
8.3 Recursive Graphs (R-Graphs): A Brief Introduction	8-7
8.3.1 R-Graphs	8-7
8.3.2 R-Operators	8-9
8.3.3 Experience with R-Graphs	8-10
8.4 Modeling Metrics Database Schema as R-Graphs	8-11
8.4.1 Mapping of Simple Schema	8-11
8.4.2 Mapping of Schema with Views	8-12
8.5 Extending R-Graph to Support Object-Oriented Abstraction and Evolution	8-13
8.5.1 Object-Oriented View Abstractions using R-Graph	8-13
8.5.2 Spatio-Temporal Modeling of Metrics Data for Predicates K and P and View Formulation	8-15
8.5.2.1 Semantic Operator S for R-Graph and A Petri-Net Based Formalism for Predicate P and Node Semantic Function	8-16
8.5.2.2 View Generation, Node Semantic and Arc Semantic Functions in R-Graph	8-17
8.5.2.3 Example of an R-Graph for Software Metrics Data	8-17
8.6 An Architecture for Software Metrics Database Management Systems	8-25
8.6.1 Summary of Findings	8-26
8.6.2 Requirements for a Software Metrics Data Model	8-27
8.6.3 Proposed Approach	8-28
8.6.4 Discussion	8-32
8.7 Conclusions	8-27

Chapter 9: Conclusion

9.1 Research Contributions	9-1
9.2 Future Extensions	9-3

Bibliography

Appendix A: Sample Data for Some Software Metrics

Acronyms	1-A
Breadth of Testing Metric	2-A
Complexity Metric	9-A
Cost Metric	55-A
Computer Resource (CPU) Utilization Metric	63-A
Computer Resource (I/O Channel) Utilization Metric	66-A
Computer Resource (Mass Storage) Utilization Metric	71-A
Depth of Testing Metric	77-A
Software Design Stability Metric	123-A
Software Fault Profiles Metric	131-A
Software Reliability Metric	142-A

Appendix B: NSDIR and NEC Data Analysis

LIST OF FIGURES	Page
Figure 1.1: Research Focus	1-3
Figure 1.2: Types of Software Metrics	1-11
Figure 1.3: Metrics Data Classification and Analysis Techniques	1-12
Figure 1.4: Data Models for Metrics Databases	1-13
Figure 2.1: Metrics Classification and Abbreviations	2-15
Figure 2.2: Sample CPU Utilization Graph for Computer Resource Utilization Metric	2-25
Figure 2.3: Graphical Display of Design Stability Metric	2-30
Figure 2.4: McCabe Cyclomatic Complexity Metric Example	2-33
Figure 2.5: Relationships Among Test and Evaluation Metrics	2-41
Figure 2.6: Conceptual Model of Use of Metrics in Software Development	2-45
Figure 2.7: Conceptual Model of Components of Software Life Cycle	2-46
Figure 2.7: Management Through Measurement	2-47
Figure 3.1: Two Aspects of Classifying Queries	3-6
Figure 3.2: Goal-Question Metric: Measurement and Evaluation Paradigm	3-7
Figure 3.3: Planned Start Date for Schedule Example	3-9
Figure 3.4: Planned End Date for Schedule Example	3-11
Figure 3.5: Actual Vs Budgeted Project Costs	3-16
Figure 3.6: Discrepancies Between Cumulative and Closed Requirements	3-17
Figure 3.7: Discrepancies Between Change Requests by User and Developer	3-18
Figure 3.8: Actual and Planned Manpower Resources	3-18
Figure 3.9: Cyclomatic Complexity of Modules for Given Project	3-20
Figure 4.1: Entity-Relation Diagram for the Proposed T&E Metrics Database	4-5
Figure 4.2: Influence Diagram for the Proposed Software Metrics	4-11
Figure 4.3: Scatterplot: Program Size versus Number of Errors	4-12
Figure 4.4(a): Distribution Plot of Effort to Design	4-15
Figure 4.4(b): Boxplot of Effort to Design	4-15
Figure 4.5: CPU Activity	4-21
Figure 4.6: MRA of CPU Activity at Level 1	4-22

Figure 4.7:	MRA of CPU Activity at Leve	4-23
Figure 4.8:	MRA of CPU Activity Metric	4-24
Figure 4.9:	Overall Process for Metrics Data Analysis	4-26
Figure 5.1:	Black Box of Classification Model	5-2
Figure 5.2:	A Simple Classification Tree	5-9
Figure 5.3:	Schematic of Neural Network-Based Classification Model	5-12
Figure 5.4:	Input Form for Neural Network-Based Classification Model	5-13
Figure 5.5:	Patterns That Are Linearly Separable	5-17
Figure 5.6:	Patterns That Are Not Linearly Separable	5-18
Figure 5.7:	The Integrated Framework for Corrective Actions and Decision Making	5-22
Figure 6.1:	Raleigh Distribution on Manpower	6-14
Figure 6.2:	Manpower Distribution Over Time When Changing Parameter	6-16
Figure 6.3:	Quality as a Function of Cost	6-19
Figure 6.4:	Requirement Stability as a Function of Cost	6-20
Figure 6.5:	Fault Profile as a Function of Cost	6-21
Figure 6.6:	Average Productivity Model Curves	6-23
Figure 6.7:	Rayleigh Distribution of Manpower Based on the Average Productivity Model	6-25
Figure 6.8:	Quality as a Function of Cost	6-27
Figure 6.9:	Requirement Stability as a Function of Cost	6-28
Figure 6.10:	Fault Profile as a Function of Cost	6-29
Figure 6.11:	Rapid Learning/Productivity Curve in Comparison to Two Linear Learning/Productivity Curves	6-30
Figure 6.12:	Average Learning/Productivity Curve in Comparison to Two Linear Learning/Productivity Curves	6-31
Figure 6.13:	Manpower Distribution of Rapid Model in Comparison to Two Linear models	6-32
Figure 6.14:	Manpower Distribution of Average Model in Comparison to Two Linear models	6-32

Figure 7.1:	Query on Planned Start Date Using the Relational Model	7-3
Figure 7.2:	Query on Schedule Slippage Using Relational Model	7-4
Figure 7.3:	Attempts at Recursive Query Using The Relational Model	7-5
Figure 7.4:	Partial Schema Definition Using Object-Oriented Data Model	7-12
Figure 7.5:	Partial Schema Definition Using Graph Data Model	7-17
Figure 7.6(a):	Object-Oriented Abstraction for Management of Heterogeneous Views	7-20
Figure 7.6(b):	Two-Level Modeling of Metrics Data	7-21
Figure 7.7:	Visual Query Primitives	7-22
Figure 7.8:	Binary Temporal Relations	7-24
Figure 7.9:	Example of Scenario Identification	7-26
Figure 7.10:	Examples of a SMPN	7-33
Figure 7.11:	Example of a SMPN Involving Embedded Non-temporal Semantics	7-34
Figure 7.12:	A Scenario Describing Various Phases in Software Development Life-Cycle	7-35
Figure 7.13:	SMPN Based Representation of Standard Database Operations	7-36
Figure 7.14:	Object-Oriented Abstraction Using SMPN	7-44
Figure 8.1:	Initial Database Schema	8-2
Figure 8.2:	Schema After Attribute Addition/Deletion	8-3
Figure 8.3:	Original Schema for Cost Metric Entity	8-3
Figure 8.4:	Adding Manpower Attribute to the Cost Metric Entity	8-4
Figure 8.5:	Schema After Adding Relationship	8-4
Figure 8.6:	Original Relationship Between Fault Profiles and Reliability Entities	8-5
Figure 8.7:	New "Time" Relationship Between Fault Profiles and Reliability Entities	8-5
Figure 8.8:	Schema After Addition of a New Entity	8-6
Figure 8.9:	A Simple Schema	8-11
Figure 8.10:	Mapping to R-Graph	8-11

Figure 8.11:	R-Graph of fault-Profiles and Reliability	8-12
Figure 8.12:	A Hierarchical Schema Involving a View	8-12
Figure 8.13:	R-Graph of Schema in Figure 8.6	8-13
Figure 8.14:	An R-Graph to Build Quality View V1	8-22
Figure 8.15:	An R-Graph to Build Quality View V2	8-23
Figure 8.16:	An R-Graph to Build Quality View V3	8-24
Figure 8.17:	An R-Graph to Support the Global Quality View V	8-25
Figure 8.18:	Proposed Architecture of Software Metrics Data Management	8-30

LIST OF TABLES	Page
Table 2.1: Elements of Cost Metric	2-20
Table 2.2: Elements of Schedule Metric	2-22
Table 3.1: Sample Queries	3-5
Table 5.1: Categories of Components In Classification Tree	5-9
Table 5.2: Data for Construction of Classification Tree	5-10
Table 5.3: Determining If A Component Is Within A Target Class	5-10
Table 5.4: Analytical Techniques For Software Metrics Data	5-21
Table 6.1: Profile of the Average Productivity Curve	6-26
Table 6.2(a): NSDIR Manpower Data Set	6-34
Table 6.2(b) NEC Data Sets (Actual Manpower)	6-36
Table 6.3: RMS Value for The Linear and Two Proposed Models	6-37
Table 7.1: Constraints with Binary Temporal Operators	7-24
Table 7.2: Query Result	7-27
Table 7.3: Buffer	7-27
Table 7.4: Interval Relation	7-28
Table 7.5: Generic Scenario	7-41
Table 7.6: Generic temporal event Abstraction of Software Metrics Data in Object-Oriented Paradigm	7-42
Table 7.7: Performance Comparison of GDM and Relational Models	7-46