Thesis written by

Amruta Hingane

B.E., Pune University, India, 2005

M.S., Kent State University, USA, 2008

Approved by

Dr. Austin Melton, Advisor

Dr. Robert Walker, Chair, Department of Computer Science

Dr. John Stalvey, Dean, College of Arts and Sciences
# TABLE OF CONTENTS

LIST OF FIGURES ...........................................................................................................VIII

LIST OF TABLES ...............................................................................................................IX

DEDICATION ................................................................................................................... X

ACKNOWLEDGEMENTS ..............................................................................................XI

CHAPTER 1 INTRODUCTION ............................................................................................ 1

1.1 Organization .............................................................................................................. 2

1.2 Terms Used ............................................................................................................... 3

CHAPTER 2 THEORY AND PHILOSOPHY OF MEASUREMENT .................................. 6

2.1 Introduction ............................................................................................................... 6

2.2 Nature and Properties of Measurement: An Informal Discussion ......................... 6

2.2.1 Properties of measurements ............................................................................. 8

2.3 The Elements Of The Formal Theory Of Measurement ......................................... 9

2.4 Quality Concept Formation .................................................................................. 11

2.5 Some Empirical Relational Systems And Direct Scales Of Measurement ............. 12

2.5.1 Extensive measurement ................................................................................... 12

2.5.2 Matching Scale ............................................................................................... 14

2.5.3 Ranking Scales ............................................................................................... 14

2.6 Indirect Measurement ............................................................................................ 15

2.7 Uniqueness: Scale Types And Meaningfulness .................................................... 15

2.8 Measurement And Other Forms Of Symbolic Representation ............................. 16
2.9 Measurement, Information And Information Machines ......................................... 17
  2.9.1 Information .................................................................................................... 17
  2.9.2 Information Machine ..................................................................................... 18
2.10 Measurement Theory In The Physical, Social And Behavioral Sciences .......... 18
  2.10.1 Measurement theory in the physical sciences ............................................... 18
  2.10.2 Measurement theory in the social and behavioral sciences ......................... 19

CHAPTER 3 MEASUREMENTS, MODELS AND SYSTEMS ................................. 20
  3.1 Introduction ............................................................................................................. 20
  3.2 Signals ..................................................................................................................... 20
  3.3 Modeling ................................................................................................................. 21
    3.3.1 Linguistic Models .......................................................................................... 21
    3.3.2 Iconic Models ............................................................................................... 21
    3.3.3 Mathematical Models .................................................................................... 22
    3.3.4 Physical Models ............................................................................................ 22
  3.4 System Structure ..................................................................................................... 23
  3.5 Development Of Mathematical Models ................................................................. 23
  3.6 The Place Of Measurements In Systems ............................................................... 24
  3.7 Models Of The Measuring Interface ...................................................................... 25
    3.7.1 The Set Theory Model ................................................................................... 25
    3.7.2 The Popular Definition Model ...................................................................... 25
    3.7.3 The Information Selection Process Model .................................................... 25
  3.8 Kinds Of Measurement Situations .......................................................................... 26
CHAPTER 4 STANDARDIZATION OF MEASUREMENT FUNDAMENTALS AND PRACTICES

4.1 Introduction

4.2 Nomenclature Of Measurement

4.2.1 Standard Nomenclature Of Measurement Science

4.2.2 Nomenclature Of A Measurement And Of Measurement Performance Of An Instrument

4.2.3 Nomenclature Describing Errors

4.2.4 Nomenclature Describing Measurement Methodology

4.3 Classification Of Measurement Science Knowledge And Practice

4.4 Units Of Physical Quantities And Their Defining Standards Apparatus

CHAPTER 5 APPLICATION OF MEASUREMENT THEORY TO SOFTWARE

5.1 Errors in software

5.2 Why standardization is missing in software?

5.3 Software tools versus physical sciences tools of measurement

5.4 Work way up to concept formation from measurement method

CHAPTER 6 GRAPH THEORY

6.1 Graph Theory Concepts

6.2 Control Flow Structure
6.2.1 Flowgraph Model of Structure ................................................................. 47

CHAPTER 7 SCALES OF MEASUREMENT ...................................................... 49
7.1 Types ............................................................................................................. 49
7.2 Summarizing the important scales ............................................................... 54
7.3 Scales with legitimate operations ................................................................. 54
7.4 Frequently Asked Questions (FAQ) .............................................................. 56

CHAPTER 8 DESIGN AND ANALYSIS OF HIERARCHICAL SOFTWARE METRICS ........................................................................................................ 57
8.1 Introduction ................................................................................................... 57
8.2 The flowgraph Model .................................................................................... 57
8.2.1 Flowgraph Definition: ............................................................................. 57
8.2.2 Cubic Graph .............................................................................................. 58
8.2.3 Cubic flowgraph ....................................................................................... 58
8.2.4 Subflowgraph Notion .............................................................................. 59
8.3 The Decomposition Theory ......................................................................... 62
8.3.1 Some conventions: .................................................................................. 64
8.3.2 Proposition ............................................................................................... 65
8.3.3 Proposition: A flowgraph is prime if and only if it’s irreducible. .......... 67
8.4 Hierarchical Software Metrics .................................................................... 71
8.4.1 Hierarchical definitions for a number of typical metrics: ..................... 71
8.5 Prime flowgraph generation ....................................................................... 73
8.5.1 Generation Method ................................................................................. 73
CHAPTER 9 CLASSICAL MEASUREMENTS AND HIERARCHICAL
SOFTWARE METRICS COMPARISON ................................................................. 74

9.1 Measurement Theory & Philosophy ............................................................. 74
  9.1.1 Quality concept formation ........................................................................ 74
  9.1.2 Uniqueness ............................................................................................... 75
  9.1.3 Metrics Correlation .................................................................................. 75
  9.1.4 Consider equalities and inequalities mentioned by Prather in hierarchical
        metrics paper [PRATHER 1995] ................................................................. 76
  9.1.5 Units ......................................................................................................... 77
  9.1.6 Questions to be asked to software metrics ................................................. 78
  9.1.7 Measurement properties in Physical Sciences and Software .................... 78
  9.1.8 Hierarchical Software Metrics and Additivity ......................................... 79

9.2 Modeling ....................................................................................................... 81
  9.2.1 System structure ...................................................................................... 82

9.3 Standardization of measurement fundamentals and practices .................... 83
  9.3.1 Standardization in software .................................................................... 84

CHAPTER 10 CONCLUSIONS ............................................................................. 88

REFERENCES ..................................................................................................... 89
LIST OF FIGURES

Figure 6.6.1 A program and its corresponding flowgraph [FENTON 1991] ................... 47
Figure 6.6.2 Some commonly occurring flowgraphs [FENTON 1991] ......................... 48
Figure 8.8.1 Splitting of a flowgraph F with respect to a proper subflowgraph F’ ......... 59
Figure 8.8.2(a) Pascal-like program text; (b) Flowgraph resulting from (a); (c)
Corresponding cubic flowgraph with process nodes; (d) Cubic flowgraph after
disregarding process nodes; (e) Underlying cubic (undirected) graph...................... 61
Figure 8.8.3 Composition G1 o G2 of two cubic graphs G1 and G2 ............................ 63
Figure 8.8.4 Degenerate cubic Flowgraphs used to represent (sequences of) elementary
process(es). ................................................................................................................ 64
Figure 8.8.5 Three flowgraphs whose sum is indistinguishable from the cubic flowgraph
of Figure 8.2(c) ........................................................................................................ 66
Figure 8.8.6 Prime Flowgraphs of orders 0, 1 and 2.................................................... 68
Figure 8.8.7 Decomposition tree for the cubic flowgraph of Figure 8.2(c).................... 70
Figure 8.8.8 Dijkstra Flowgraphs ............................................................................. 71
Figure 9.1 Direct comparison using subgraph methodologies.................................. 86
LIST OF TABLES

Table 7.1 Some important scales [FENTON 1991], [FENTON 1997, [WIKIPEDIA 2008]

................................................................................................................................... 54
DEDICATION

A POT of Software Metrics

A Physiological Overtum of Technology of Software Metrics

History of the title: A sunny afternoon, I was discussing the topic for research with a friend. The discussion spanning over an array of topics landed into a debate on how the software metrics be standardized. If only there was a pot, in which we could place all the metrics together and do a scrutiny over each, trying to understand how they could relate to

- one another
- this world, and
- the metrics used to measure the physics of this world;

…it would serve, to great extent, the purpose of this research. Thus from the womb of that discussion, was born the title of this theses report…

A POT of Software Metrics

(A Physiological Overtum of Technology of Software Metrics)

Thence I dedicate the title to that friend- Mr. Anand Sakhrani and this study to Dr. Austin Melton and all those who in some way lead to shaping of this research.

Thanks!
ACKNOWLEDGEMENTS

I acknowledge all those who helped and supported me in finishing this thesis.

The constant support and encouragement I received from my academic advisor, Dr. Austin Melton helped make this thesis possible. From the first vague proposal of this topic to later queries on focus and connection, he was always eager to entertain my ideas and help me solve conceptual difficulties. I sincerely thank him for his consistent efforts and true desire to keep me on track.

I would also like to thank my committee members, for serving in my defense committee despite their overwhelmingly busy schedule.

Thanks to the staff at the Kent State Department of Computer Science, in particular Marcy Curtiss who helped and encouraged me every step of the way during my time in the department.

Last but not the least; I would like to express my deepest gratitude to Anand Sakhrani and my parents. Their support and unwavering confidence in my ability helped me achieve my academic dreams.

Amruta Hingane

August 19 2008, Kent, Ohio
CHAPTER 1

Introduction

A measure is an empirical objective assignment of a number (or symbol) to an entity to characterize a specific attribute. Such an assignment of numbers is also referred to as measurement mapping. Metrics are essential in all sciences. The terms metric and measurement are often used interchangeably in this thesis.

Measurement is required in many systems that govern our lives. We use it to calculate the bill total in order to see whether we get back the correct change at a shop. We measure the height of children to make sure we get the right size dress. We work out how far will we be traveling when we set out on a journey to predict how far will the journey be or how much petrol will be required. Measurement is concerned with capturing information about attributes of entities. It assigns numbers or symbols to attributes of entities in order to describe them.

Software metrics is a term that includes models and measures of widely ranging activities such as cost and estimation, productivity, quality control and assurance, data collection, reliability, performance evaluation, algorithmic/computational complexity, structural metrics and so on. But measurement is often neglected in software engineering. We still fail to set measurable targets while developing software products. We fail to measure various components that make up the real costs of software projects. We cannot tell the potential user how reliable a product will be in terms of likelihood of failure in a given period of time. In theory we have many kinds of metrics for measuring different
software attributes. A subset of these is hierarchical software metrics based on hierarchical decomposition structures of underlying flowgraphs. These metrics use sequencing and nesting operations for decomposition.

Do software metrics follow the properties of metrics in physical sciences? If they are different, in what respect they differ is what we will see in the following chapters. We focus on hierarchical software metrics to carry out this comparison.

1.1 Organization

Chapter 2 gives us a summary of metrics in physical sciences. It talks about the fundamentals, properties, conditions and concepts involved in classical metrics. Chapter 3 summarizes the modeling theory that is important in devising new measurements and employing them. Chapter 4 gives us the standardization theory that governs measurement in physical sciences. It tells us why the standardization of measurement fundamentals and practices is required and how to perform them. Chapter 5 discusses the application of measurement theory to software measurement in terms of errors, measurement tools and standardization. Then in Chapter 6, we see an introduction to graph theory, the basic concepts used and the control flow structure. Chapter 7 gives us an overview of scales and its types used in measurement theory. Chapter 8 summarizes the hierarchical software metrics and the concepts, notions and decomposition methods involved in them. Chapter 9 gives us the comparison between properties, concepts and methods of measurements in the physical sciences and those in software. We will see the differences
and similarities and changes that need to be made for the two types of metrics to follow the same strategies and to standardize them.

1.2 Terms Used

1. Relations

A relation R between elements of sets A and B (sets are denoted by italicized letters) is a subset of \( A \times B \); \( R \subseteq A \times B \). A relational system \(<A, R>\) is a family of sets consisting of a set and a class of relations R defined on it.

2. Special Types of Relations

a) R is called symmetric if and only if whenever \((a, b) \in R, (b, a) \in R\).

b) R is called reflexive if and only if for each \(a \in A\), \((a, a) \in R\).

c) R is called transitive if and only if whenever \((a, b) \in R\) and \((b, c) \in R\) then \((a, c) \in R\).

d) A relation R is called connected in \(A\) if for all \(a, b \in A, a \neq b\), either \((a, b) \in R\) or \((b, a) \in R\).

3. Equivalence, Order

A relation R that is reflexive, symmetric and transitive is called an equivalence relation. It will be denoted by ~.
An equivalence relation \( \sim \), is a congruence relation for a relational system \( <A, \mathcal{R}> \) if whenever \( a \sim b \), a can be substituted for b in any \( R \in \mathcal{R} \). This is called the substitution property of the relation.

Equality, \( = \), is an equivalence relation between mathematical entities, which has the substitution property with respect to all relational systems for those entities.

A relational system \( <A, \sim, < > \) will be called an order system if and only if:

a) \( a, b \in A \) exactly one of the relations holds \( a \sim b, a < b, b < a \);

b) \( \sim \) is an equivalence relation;

c) \( < \) is transitive.

An example of an order system is the numerical system \( <\mathbb{R}, =, < > \).

4. Homomorphism

Given two relational systems \( <A, \mathcal{R}> \) and \( <B, \mathcal{P}> \); where \( A \) and \( B \) are sets and \( \mathcal{R} \) and \( \mathcal{P} \) are classes of relations on \( A \) and \( B \) respectively, \( \mathcal{R} = \{R_1, ..., R_n\} \) and \( \mathcal{P} = \{P_1, ..., P_n\} \) consider that there is a mapping \( M \) from \( A \) onto (into) \( B \), \( M: A \rightarrow B \).

Then \( M \) is a homomorphism from \( <A, \mathcal{R}> \) onto (into) \( <B, \mathcal{P}> \) if and only if for all \( a_1, ..., a_k \in A \).

\( R_i(a_1, ..., a_k) \Leftrightarrow P_i(M(a_1), ..., M(a_k)) \) for \( i = 1, ..., n \) (\( \Leftrightarrow \) means implies and is implied by).
5. Operations

Let $A$ denote any set. Then a binary operation $\circ$ on $A$ is a mapping which assigns to each pair of elements $a, b \in A$ and element $a \circ b \in A$.

An operation is commutative if:

$$a \circ b = b \circ a$$

An operation is associative if:

$$a \circ (b \circ c) = (a \circ b) \circ c$$
CHAPTER 2
Theory and Philosophy of Measurement

2.1 Introduction

This chapter gives us the properties of measurements, basic concepts, some measurement fundamentals and measurement kinds. This chapter is a summarization of chapter 1 of Sydenham [SYDENHAM 1982].

Measurement is the process of empirical, objective assignment of numbers to the properties of objects and events of the real world in such a way as to describe them. Measurement of what is observed is the goal towards which scientific investigation is directed. Galileo Galiliei made the statement “Count what is countable, measure what is measurable; and what is not measurable, make measurable”. Measurement enables the laws and theories of science to be expressed in the precise and concise language of mathematics. When a number characterizes the property of an object or event this number carries information about the property.

2.2 Nature and Properties of Measurement: An Informal Discussion

1. Firstly, measurement is the assignment of numbers to properties of objects or events and not to objects or events.

2. The definition “Measurement is the process of empirical, objective assignment of numbers to the properties of objects and events of the real world in such a
way as to describe them”, states that the assignment of numbers in measurement is such that the numbers describe the properties of the object or event.

3. The meaning can be explained as follows. Consider that a number, or measure is assigned by measurement to the property of an object and the same process assigns other numbers to other manifestation of the property. Then the numerical relations between the numbers or measures imply and are implied by empirical relations between the property manifestations. Thus if numbers assigned to the manifestations of a particular property in two objects by measurement are equal; this implies that the two property manifestations are empirically indistinguishable. Conversely empirical indistinguishability implies the equality of measures. Again if the numbers assigned by measurement to the manifestations of a particular property, in a series of objects, can be placed in order of increasing magnitude, this implies that there is an empirical relation which would result in the placing of the objects in the same order in respect of the property, Conversely, an empirical order among manifestations of the property, implies the same order among the measures.

The above clearly indicates that measurement is a process of comparison of a manifestation of a property with other manifestations of the same property. Many definitions further state that the measure of a property expresses the ratio of the magnitude of the property to a standard magnitude taken as unity. This begs the essential
question of what measurement is. The statement is untrue for many scales of measurement and would make the measurement of many properties impossible.

1. For example body temperature on the Celsius scale is not the ratio of the temperature to a unit degree Celsius.

2. Another example is intelligence of a person that cannot be measured as a ratio, to the intelligence of a person having unit intelligence.

There is a divergence of views as to whether any descriptive assignment of numbers is adequate for the process to qualify as measurement.

1. At one extreme, broadly in the social and behavioral sciences, there is the view that any empirical, objective assignment of numbers that describes a property manifestation can be termed measurement.

2. At other extreme, it is the view that only numbers that reflect in some way a ratio to a unit magnitude of a property are true measures. This is the classical view and true for most informal definitions in physics.

2.2.1 Properties of measurements

1. A measurement is an objective description and hence a proper scientific datum. Conversely it can be claimed that if we can arrive at a totally objective description of a property manifestation, the most vital step towards measurement has been taken.

2. Measurements are descriptions of great conciseness. A single number tells us what it would take many words to express.
3. A measure of a property gives us an ability to express facts and conventions about it in the formal language of mathematics.

2.3 The Elements Of The Formal Theory Of Measurement

The representational theory of measurement has four parts:

1. An empirical relational system corresponding to a quality (or attribute)
2. A number relational system
3. A representation condition
4. A uniqueness condition.

1. Quality as an empirical relational system

For some quality:

\[ Q = \{q_1, q_2, \ldots, q_i, \ldots\} \] is the set of all possible manifestations of the quality

\[ \Omega = \{w_1, w_2, \ldots, w_i, \ldots\} \] represents the class of all objects manifesting elements of Q such that \( q_1 \) is for \( w_1 \), \( q_2 \) is for \( w_2 \) and so on.

\[ R = \{R_1, R_2, \ldots, R_i, \ldots, R_n\} \] is the set of empirical relations on Q

Thus quality is represented by empirical relational system

\[ Q = \langle Q, R \rangle \]

2. Numerical relational system

\[ P = \{P_1, P_2, \ldots, P_i, \ldots, P_n\} \] is the set of relations defined on N which is a class of numbers
The numerical relational system is represented by
\[ \mathcal{N} = \{N, P\} \]

3. Representation condition

Representation conditions require that the relations between the referent property manifestations imply and are implied by the relations between their images in the number set. Formally measurement is defined as an objective empirical operation

\[ M : Q \rightarrow N \]

\[ Q = \langle Q, R \rangle \] is mapped homomorphically into (onto) \[ \mathcal{N} = \langle N, P \rangle \] by \( M \) and \( F \). \( F \) is one to one mapping.

\[ F : R \rightarrow P \]

…such that \( P_i = F (R_i) \); \( P_i \in P \); \( R_i \in R \)

\( P \) is an n-ry relation if and only if it is the image under \( F \) of an n-ry relation.

Measurement is homomorphism because \( M \) is not one-to-one. It maps separate but indistinguishable property manifestations to the same number. Then

\[ Y = \langle Q, \mathcal{N}, M, F \rangle \]

…constitutes a scale of measurement for \( n_i = M (q_i) \). The image of \( q_i \) in \( N \) under \( M \) is called the measure of \( q_i \) on scale \( Y \).

4. Uniqueness Condition

The representation condition may be valid for more than one mapping \( M \). One may admit certain transformations from one scale of a property to another without
invalidating the representation conditions. The uniqueness condition defines the class of
scale transformations to those for which the representation condition is valid.

2.4 Quality Concept Formation

1. Both in the historical development and logical structure of scientific knowledge,
   the formulation of a theoretical concept or construct, that defines a quality,
   precedes the development of measurement procedures and scales.

2. Example, the concept of hotness as a theoretical construct interpreting the
   multitude of phenomena involving warmth is necessary before one can conceive
   and construct a thermometer. Similarly, hardness must be defined before we
   establish a scale for its measurement.

3. The concept of a quality is formed as an objective rule for the classification of a
   collection of empirically observable aspects of objects into a single set, together
   with the family of objective empirical relations on that set. The resulting
   relational system is a quality and each single member of the set is termed a
   manifestation of the quality.

4. We can thus see that there is a difficulty in the measurement of such qualities as
   beauty. The existence and meaningful use of the word beauty indicates the
   usefulness of the concept. However, there is not an objective rule for classifying
   some aspect of observable objects as manifestations of beauty. Similarly there are
   no objective empirical relations such as indistinguishability or precedence in
respect of beauty. The basis for measurement for beauty is thus absent from the outset.

5. In some cases the concept of a quality arises from invariance in numerical laws arrived at by measurement.

6. In general however, one starts from some direct concept of a quality and then seeks measurement scales.

### 2.5 Some Empirical Relational Systems And Direct Scales Of Measurement

#### 2.5.1 Extensive measurement

Extensive measurement is measurement of physical quantities for which we can construct an operation having the formal properties of addition as the basis of physical measurement.

1. The extensive scales of physical measurement are based on establishing for the quality \( Q \) of empirical objects, for which a scale is to be determined of an empirical ordering with respect to \( Q \) of the class \( \Omega \) of all objects, elements of \( \Omega \), which has with respect to \( Q \) the formal properties of addition. Such scales are known as extensive. The basis of a scale of measurement of \( Q \) is the definition of the set \( Q \).

2. Secondly, there must be an operational procedure that establishes on the set of objects \( \Omega \) possessing \( Q \) an empirical equivalence relation \( \sim \) and a transitive empirical relation \( < \) with respect to \( Q \) such that \( <Q, \sim, < > \) in an order system.
3. Consider objects \( w_1, w_2, w_3, w_4 \in \Omega \) exhibiting property manifestations \( q_1, q_2, q_3, q_4 \in Q \) respectively. For an extensive measurement scale there must be an operation \( \circ \) of combining objects, and then combining objects must be respected by the measurement. We also use \( \circ \) to denote the operation on the measurement values, e.g., \( q_1 \circ q_1 \) denotes the measurement of \( w_1 \circ w_2 \). With respect to the measurement \( \circ \) acts like addition.

For all \( q \in Q \)

a) \( q \circ q_2 \in Q \)

b) \( q_1 \circ q_1 \sim q_2 \)

c) \( q_1 \circ q_2 \sim q_2 \circ q_1 \) \quad \text{commutativity}

d) \( q_1 \circ (q_2 \circ q_3) \sim (q_1 \circ q_2) \circ q_3 \) \quad \text{associativity}

e) if \( q_2 \sim q_3 \) then \( q_1 \circ q_2 \sim q_1 \circ q_3 \)

if \( q_3 \sim q_2 \) then \( q_1 \circ q_3 \sim q_1 \circ q_2 \)

f) if \( q_1, q_2, q_3, \ldots \) bear to each other the relation \( \sim \) and \( q_1 < q'_1 \), then there is a number \( n \) such that \( q'_1 < q_1 \circ q_2 \circ \ldots \circ q_n \) \quad \text{Archimedean postulate}

With these definitions the empirical relational system \( \langle Q, \sim, <, \circ \rangle \) has a structure equivalent to the numerical relational system \( \langle \mathbb{R}, =, <, + \rangle \).

Example: In the measurement of mass the equipoise balance offers the means of establishing empirical order. If the arm balances, the masses in the two pans are equivalent. The tipping of the balance indicates that one mass is heavier than the other.
Thus we rank a series of weights in order of heaviness. The lumping together of two objects is with respect to mass an operation with the properties of addition.

2.5.2 Matching Scale

1. Given \(<Q, \sim>\) a set of differing elements

\[ s_i \in Q \ (s_i \sim \ s_j \text{ if } i \neq j) \]

are selected to form a standard set \( S = \{ s_1, s_2, \ldots, s_k \} \)

Numbers (or other symbols) \( n_i \in N \) are then assigned to each \( s_i \in S \) (if \( s_i \sim s_j, n_i \neq n_j \))

The fundamental measurement operation \( M \) consists of an empirical operation in which measurands \( q_i \in Q \) are compared with members of the standard \( S \).

If \( q_n \sim s_i \) it is assigned the number \( n_i \).

2. Example: A color code in which the relation ‘matches’ constitutes the empirical indifference relation. This relation is based on a color match. It can be tested for symmetry and transitivity; reflexivity is implicit. Thus the relation is an equivalence relation.

2.5.3 Ranking Scales

1. In ranking scales, an empirical order system \(<Q, \sim, \rangle\) is established on \( Q \), a set of differing standard objects having \( s_i \in Q \) is then selected an arranged in an ordered standard series \( S = \{ s_1, \ldots, s_n \} \) according to \(<Q, \sim, \rangle\). Numerals are assigned to each \( s_i \) say \( i \) in such a way that the order of numerals corresponds to
the order in $S$ of standards to which they are assigned. Any $q \in Q$ can then be compared with the elements of $S$ in the same way as in nominal measurement. If $q$ bears the relation $\sim$ to any $s_i \in S$ it is then assigned the numeral of $s_i$. If an entity is not equivalent to any $s_i \in S$ one can determine between which two standard elements it lies in the empirical order system.

2. Example: Mohs scale of hardness of minerals.

2.6 Indirect Measurement

1. A case when every object that manifests the quality to be measured exhibits a set of other qualities that are measurable; then to each a manifestation of the measurands quality there corresponds a set of measures of the associated qualities. These associated or component measures can be arranged in an ordered array.

2. If manifestations of the measurand quality have identical arrays of component measures, and if and only if they are indistinguishable, then the array of component measures characterizes the measurand.

2.7 Uniqueness: Scale Types And Meaningfulness

1. The class of transformations that transform one member of a class of equivalent scales into another is called the class of admissible transformations.
2. The conditions that admissible transformations must satisfy are known as uniqueness conditions. They specify that the scale is unique up to a specified transformation.

3. Only those statements involving measurements are meaningful that can be logically traced to the empirical operations on which the measurement is founded.

4. A statement made about a quality in terms of its measures is meaningful if its truth is unchanged by admissible transformations of the scales of measurement; in other words, if it reflects the empirical relational system on which the scale is based and not just the arbitrary conventions of the scale.

2.8 Measurement And Other Forms Of Symbolic Representation

1. Measurement is only one form of representation of entities by symbols.

2. It is closely related to other forms of symbolization. At a simpler level, a symbol may only be a name or a label that can be used to refer to the object and handle information about it.

Theory of foundations of measurement can be extended to embrace forms of representation of entities by symbol systems other than numerical ones. The basic concepts of foundational theory of measurement, such as representation, uniqueness, meaningfulness, etc, can be extended to general symbolic systems.
Non-numerical representations are like measures that describe the entity represented and some of its relations in a form that enabled information about the entity to be conveniently manipulated.

The most important special feature of measurement is that the assignment of numbers or symbols is objective and represents empirical facts or empirical observations.

2.9 Measurement, Information And Information Machines

2.9.1 Information

1. Information consists of the symbol together with the relation it bears to the referent. If we have a referent relational system \( Q = \langle Q, R \rangle \) a symbolic relational system \( Z = \langle Z, P \rangle \) and mappings \( M: Q \rightarrow Z \) and \( F: R \rightarrow P \) giving us a code \( C = \langle Q, Z, M, F \rangle \), then \( I = \langle C, z_i \rangle \) represents information about a \( q_i \) for which \( z_i \) is a symbol and about relations between \( q_i \) and other members of \( Q \).

2. This ‘information’ differs from that of information theory with essential similarities.

3. In information theory, an information transmission channel is considered that transforms element \( x_i \) of a set of inputs \( X \) into elements of a set of outputs \( Y \) that is in general a many-to-many transformation.

4. If we consider \( Q \) as analogous to \( X \), \( S \) as analogous to \( Y \) and \( M \) as analogous to the \( X \) to \( Y \) transformation of the communication channel then the underlying concepts seem to be fundamentally similar.
5. In both, information is knowledge about and entity provided by an image of the entity under a mapping.

2.9.2 Information Machine

1. An information machine functions by performing a prescribed transformation of an input physical signal to an output physical signal.

2.10 Measurement Theory In The Physical, Social And Behavioral Sciences

2.10.1 Measurement theory in the physical sciences

1. Measurement in physical sciences is based on establishment of direct extensive scales of measurement for a number of physical quantities, which are used as the base of a system.

2. Scales for other physical quantities are obtained as derived scales, that is indirect scales in terms of the base quantities, in the form of multiplicative monomial functions of the base quantities.

3. For example, the SI (The International System of Units) system of units has seven base quantities: length, mass, time, electric current, thermodynamic temperature, amount of substance and luminous intensity. The base unit of current for example is defined as a current maintained in two parallel conductors of infinite length, negligible circular cross section, and placed a specified distance apart, produces a specified force per unit length.
This firstly involves scales of measurement of length and force. Then it involves a
theory of electromechanical interaction that is a law of force between conductors. All this
is essential to enable us to calculate the instrument law of a current balance by which the
unit of current must be realized.

The actual realization of the definition in terms of infinitely long infinitesimal
conductors is not possible. Even for quantities such as length, for which an extensive
scale can be established using concatenation of standard lengths, the unit of light is in fact
defined in terms of wavelength of light, which is not a material object but a construct of
physical theory.

2.10.2 Measurement theory in the social and behavioral sciences

1. The social and behavioral sciences are more concerned with attributes or qualities
   such as standard of living, utility, alienation, intelligence, etc.
2. The first attempt to measure them has the difficulty of establishing an adequate
   objective concept of these qualities based on empirical operations.
3. The conceptual framework is often absent.
4. In these sciences, it is by no means universally agreed that the clear formation of
   concepts in terms of empirical observation is possible or desirable; nor is there an
   agreement that the search for data, through measurement, advances knowledge
   and understanding.
5. The opponents of quantification would say that human nature and behavior are
too variable to enable the methodologies of the physical sciences to be applicable
to them.
CHAPTER 3
Measurements, Models And Systems

3.1 Introduction

It is necessary to understand the measurement situations in order to employ an appropriate measurement strategy. Efficient design and application of hardware requires of the user a broad understanding of the role that a measurement hardware piece plays in the total system that it is placed in. In practice that situations in the hard-sciences generally are relatively straightforward to realize conceptually because the parameters to be sensed are clear-cut. In contrast is the situation of soft sciences can be such that it is not possible to decide what to be measured or how to do it if identified. This chapter gives us the summary of the modeling theory involved in measurements from Chapter 2 of Sydenham [SYDENHAM 1982].

3.2 Signals

1. Measurement information is conveyed in some suitable form via an energy or mass transfer link, the measurement information being conveyed by such a carrier as changing state or modulation of the carrier.

2. The changing medium used in this way is termed the signal.

3. If the modulation conveys no useful or desired information it is termed as noise which if excessive can mask the true signal providing false information.
4. The energy or mass medium carrying the signal is embodied in the system instrumentation.

3.3 Modeling

1. Models provide representation of some aspect of the real-world system of interest.
2. They enable us to investigate the real situation without needing to actually produce it or modify an existing situation.
3. They provide means to motor ingenuity that is they give us a chance to imagine the working of the real system.

3.3.1 Linguistic Models

1. Theses models use the natural spoken language to express sufficient parameters and their interactions, of the system of interest such that an adequate level of explanation of information transfer takes place.
2. In empirical sciences and arts, this is the prime form for presenting models of situations and relationships.
3. It can rarely serve hard sciences and has been adopted and used in soft sciences.

3.3.2 Iconic Models

1. An icon is an image, figure or pictorial representation of a concept.
2. Iconography is the description of a subject by means of drawings or figures.
3. In engineering and sciences iconography is used extensively.
4. Example: Three-view engineering drawings of objects, circuit diagrams, block schematic diagrams, isometric projection drawings and graphs of various kinds.

5. The iconic level of modeling is usually very early reached in any design or discussion of a measurement system problem.

3.3.3 Mathematical Models

1. Mathematics is in itself a modeling discipline for it enables relationships between defined quantities to be expressed in terms of representative mathematical symbols and statements.

2. In many areas this form of models expresses system fundamentals in conceptual depth well beyond its realization as practical systems.

3. Mathematical models are able to run faster in time than the physical system they represent.

4. They enable a degree of prediction.

5. A subject can be considered to have reached a level of maturity if it can be modeled mathematically to an adequate level.

3.3.4 Physical Models

1. It is often convenient to construct physical models of a situation.

2. Example: Scaled down versions of a proposed process plant or a model of a device seeking patent approval.
3.4 **System Structure**

Systems can be reticulated or box-cut, a process in which the whole is, for purpose of analysis, broken down into more detailed assemblies of subsystem blocks, the process being taken as far as is needed, perhaps finally to the basic elements.

3.5 **Development Of Mathematical Models**

It is valuable to describe the steps taken in treatment of a topic where this form of model is sought.

1. First step is to consider the subject to be modeled from any and every aspect that appears relevant.
2. As features, aspects, relationships are realized they are recorded generally as symbolic black boxes with at least a linguistic form of description.
3. Second step is to sort the general information into systematic classes aiming to identify the various black boxes and their interconnections.
4. This iconograph is then continually refined and broken down until it is possible to see the individual at such a level of simplicity that they can be assigned appropriate general mathematical input-output relationships.
5. Linear equations are preferred but non-linear expressions can also be handled.
6. In many cases the arrangement of boxes and the equation selected as the model may only provide the overall transfer function sought with internal operation being quite different at the nodes (or ports) of the system.
7. Such a model is relevant to a wide range of physical manifestations so the numerical coefficients of the various equations selected need to be identified.

8. It is necessary to convert all such parameters that are measurement variables, into the numerical state to bind the model into an adequate specific one.

9. Models must be proven, evaluated in some way to ensure that they do indeed represent the reality desired.

10. Usually the first prepared models are proven inadequate in some respect and the process of refinement continues until one runs out of ideas, effort, or more optimistically, until the model provides an adequate level of simulation.

3.6 The Place Of Measurements In Systems

1. A key purpose of measurement is to map a physical parameter into an equivalent number set.

2. A measuring system comprises a sensing stage, in which the original parameter to be measured (measurand) is transduced into an appropriate equivalent signal.

3. The sensor’s role is to extract specific information, to act as an information filter, passing information on the state of a particular chosen parameter existing within a possibly infinite set of definable parameters that totally describe the system.

4. A measuring system, along with conveying internal messages (signals) without loss of accuracy, should also, overall, map variables in a faithful manner.
3.7 Models Of The Measuring Interface

3.7.1 The Set Theory Model

1. A basic rigorous mathematical model for what occurs at the interface formed between a sensor and the system to which it is connected is based on set theoretical considerations.

2. This model of measurement appears to be most fundamental and being mathematical, it should ultimately pave its way to machine-decision making about the design of a measurement interface.

3.7.2 The Popular Definition Model

1. Measurement is comparing an unknown quantity against a defined standard for that kind of quantity that can be embodied in some form of sub-divisional scale.

2. Similar to set-theoretical model, this model also indicates the need for an agreed standard unit for the parameter and a way to realize it.

3. It also makes apparent the need for a method by which the standard and the unknown are compared.

4. Also revealed is the problem of what to do about any difference between a standard and unknown that lie outside exact integer equality.

3.7.3 The Information Selection Process Model

1. Information in common language refers to facts, ideas, entities, concepts and attributes that define a subject or object.
2. In information theory sense information is concerned with the quantity conveyed in a message passing through a communication channel.

3. In this sense the conceptual idea of meaning of that message is not catered for by the theory.

4. An apparently nonsensical message can be transmitted with utmost fidelity.

5. Thus a measuring system is concerned with both kinds of information. It must map the variable (that is codify the measurand) and also transmit it according to information theory.

3.8 Kinds Of Measurement Situations

3.8.1 Interaction Between The System And The Measuring Stage

The measuring interface comprises the system of interest and some kind of measurement sensor that codes meaning to what is sensed.

Four classes of this can be identified as an aid to understanding what might take place. The basis of the class into which to place a specific situation is based on the rigidity of the code associated with the meaning allocated.

1. Class 1. This is the case when the designer has created a hardware sensor and applied it to produce data to which a clearly defined meaning has been stated. The code is rigidly applied; it is thereafter assumed until recalibration that the code remains unchanged. Clearly there is a danger that malfunction or an additional
noise source could occur altering the code. So many measurement systems will have a periodic self-checking feature.

2. Class 2. The process is unaltered by the application of a properly designed sensor but here the observer changes it as observation continues. In many cases the knowledge obtained by the act of measurement alters the observer’s concept of what is being measured. The meaning that is ascribed by the user in the first place, can unwittingly be changed as measurement proceeds. This situation arises in the use of measurements to seek new knowledge.

3. Class 3. When two or more observers communicate using their natural senses and data processing ability the situation of class 2 extends to both systems possibly changing as observation of each continues.

4. Class 4. The logical extension of class 3 is to the state where two men created hardware intelligent systems sense each other and act accordingly.

3.8.2 Access To System Measurement Nodes

1. Accessible real parameters: Of potentially infinite parameters that might need to be measured only some of them will, at any given time and state-of-the-art be accessible.

2. Inaccessible real parameters: Although it can be reasoned or by indirect method experimentally shown that real parameters exist that would be worth measuring, in some cases it is not possible to do so for practical reasons. For example, measurement of many human physiology parameters is not yet possible under
normal operating state conditions, as measurement methods would cause permanent damage.

3. Inaccessible unreal parameters: Models can generate state parameters that have no real physical relationships to the system modeled. Thus any amount of direct measurement effort will not lead to measurements being made. The data sought, however, might be obtained by indirect method.
CHAPTER 4

Standardization Of Measurement Fundamentals And Practices

4.1 Introduction

The basic concepts of measurements are applied in many different disciplines in many different ways. Putting them to general practical use usually requires the establishment of agreed procedures that are controlled by the use of agreed primary standards of various defined units. This in turn creates a need for standardized nomenclature to describe the concepts concerned. This chapter establishes an understanding of standard procedures, practices and concepts in the areas of nomenclature, physical standards and standards of specification. It summarizes chapter 3 of Sydenham [SYDENHAM 1982].

4.2 Nomenclature Of Measurement

4.2.1 Standard Nomenclature Of Measurement Science

1. Measurement science is often referred to as metrology this being the field of knowledge concerned with measurement.

2. A problem of use of various synonyms to describe the same, approximately equal, concepts exist.
4.2.2 Nomenclature Of A Measurement And Of Measurement Performance Of An Instrument

There are three distinctly different concepts in conducting measurement or describing the performance of a measuring instrument.

1. The first relates to the resolving ability of a measuring process called discrimination. It is the quality that characterizes the ability of the measuring instrument to react to small changes of the quantity measured.

2. The quality that characterizes the ability of a measuring instrument to give the same value of the quantity measured, not taking into consideration the systematic errors associated with variations of the indications is defined as repeatability.
   
   a) It is important to distinguish between the closeness of values that define the repeatability of the individual, or the group of values, when measured in the short term with the same apparatus and the same parameter determined by a long term set of measurements or by different persons with different apparatus. The latter case is called reproducibility.
   
   b) When the instrument, the observer’s performance and the external perturbing parameters are each constant, variation in the values being caused by changes of the measurand itself is called repeatability of the measurement rather than of the instrument.

3. Accuracy of an instrument is the quality that characterizes the ability of a measuring instrument to give indications approximating to the true value of the
quantity measured. It is an expression of the truthfulness available: lack of accuracy arises from both the instrument and the imperfect standard of the unit.

a) Measurement is a standard of understanding errors as much as measurements.

b) Accuracy is finally assigned to an instrument by agreement: it does not automatically arise from good design alone.

### 4.2.3 Nomenclature Describing Errors

1. Measurements are never perfect; errors occur as deviations from the perfect case. Two dominant general groups of errors are systematic and random.

2. The range within which the true value lies is termed the uncertainty.

3. Systematic errors are those that can be predicted from past knowledge of the operation involved, on an individual measurement basis.

4. Random errors are those that cannot be predicted on an individual basis but for which a statistical method can yield information about the mean value of a set of data using the theoretical laws of probability.

### 4.2.4 Nomenclature Describing Measurement Methodology

The methodology of making measurements also has a defined terminology that enables efficient communication of concepts by the use of generally accepted terms. All measurements, although, are made by comparing measurands against a defined standard in some way, there exist many ways to achieve this.
1. In direct method the value of the quantity to be measured is obtained directly, without the necessity of supplementary calculations based upon a functional relation between the quantity to be measured and other quantities actually measured.

2. An indirect method of measurement is that in which the parameter sought is gained by use of intermediate stages of different units that are linked.

3. Comparison method of measurement in a method based upon comparison of the value of a quantity to be measured with a known value of the same quantity, or with a known value of another quantity, which is a function of the quantity to be measured.

4.3 Classification Of Measurement Science Knowledge And Practice

The knowledge of measurement science is scattered over many areas of its application, of which only little is found in few specialist groups devoted to its fundamentals. So it is necessary to search for terminology in various similar applications as well as the class in which general terms are covered.

1. For example, there exist many standards about nomenclature in fields such as acoustic noise, temperature measurement and flow metering.

2. Nomenclatures of these overlap considerably in concepts, different words for the same concept are often used.

3. Thus there is often a large and confusing array of terms issued in several standards on the same topic.
The internal taxonomy of knowledge and its terminology for measurement science are just beginning to become organized.

**4.4 Units Of Physical Quantities And Their Defining Standards Apparatus**

The common concept that is most close to the actual working face of the application of measurements is that a measurand is compared against the defined standard, or something representing that standard, the difference from the actual magnitude of the standard being expressed in subdivisions of the basic unit used according to some form of scaling.

1. The standard is the physical object or characteristic of a physical apparatus that represents the conceptual unit chosen to represent a particular measurable attribute.

2. For example, a particular piece of metal uniquely represents the unit of mass; here the unit is kilogram. It is represented by a sole piece of material maintained under defined and controlled conditions and is considered to be the physical standard of the unit of measurement.

The general philosophy adopted for the creation of primary standards is that they be based upon some physical principle that is known to be as invariant as can possibly be found. It is more expedient to retain any new standard’s value close to the magnitude of unit adopted than it is to change the whole, traceable chain of units whenever a new and better physical principle is discovered.
Most of the primary standards are now based upon natural physical principles but their units are still declared as man chooses.

1. Their values are refined as better knowledge is gained.

2. Development and maintenance of the primary physical standards is a specialist area of measurement science.

3. Primary task is to develop, maintain and apply apparatus that will provide definitions of the required base and derived units at the highest possible accuracy and reproducibility.

4. Factors such as cost, time to set up and make an observation, size and portability of the apparatus, ability to be mass produced for commercial scale and other factors important to the industrial user, are of lesser significance in this case than achieving the best possible metrological performance. In each extreme the basic philosophy is much the same, it is the emphasis that differs.

5. It takes many years to develop new standard apparatus and obtain agreement for its general use throughout the cooperating countries. Nomenclatures for the physical standards have not yet achieved a single uniform terminology.
CHAPTER 5

Application of Measurement Theory to Software

Software measurement is relatively a new field compared to measurements in physical sciences. Software measurement needs a theory so that we can know what we are doing and whether or not we are making progress. How do we verify what we are doing? This chapter compares properties and concepts in measurement theory with software measurement and tries to apply them wherever possible.

5.1 Errors in software

In regard to historical measurement i.e. measurement in physical sciences, Sydenham [SYDENHAM 1982] says that measurements are never perfect. Errors occur as deviations from the perfect case (the expected value of measurement). In measurement theory, two dominant general groups of errors exist called systematic and random. The range within which the true value lies is called uncertainty. Systematic errors are those that can be predicted from past knowledge of the operation involved, on an individual measurement basis. Random errors are those that cannot be predicted on an individual basis but for which a statistical method can yield information about the value of a set of data using the theoretical laws of probability. For example, measuring the inner diameter of an iron pipe of certain thickness, during summer provides a certain measurement value. The same pipe’s diameter when measured during winter can show a different value because iron gets affected due to temperature changes. So, this can be considered a
random error in the measurement. There could be many more such errors possible in measurement theory for physical sciences.

Does software measurement consider such errors? What errors can arise when we perform software measurements? One could make errors like measuring a line of code less while counting, an undeclared variable is not noted while performing measurement which may lead to improper functioning of the code, different kinds of loops (if-else, do-while, etc.) are given same values, an operator gets considered as an operand or an operand is considered to be an operator and other such errors. Different modules of a system work fine on their own but after their integration the performance may vary from the perception or their functioning may deviate from the perfect case. Tools used for performing software measurement may have certain errors too which need to be removed so that the measurement obtains correct values. It could be possible that the value obtained by measurement and the desired one are close enough for the deviation to be allowed so as to cut on the costs and effort that could be involved in correction of the errors. Environmental factors do not affect software, hence they cannot affect software measurement the way they affect measurement in physical sciences. So software measurement has a property: Software measurement does not have random errors.

Do all the classes of errors found in measurements in physical sciences apply to software measurements too? It is possible that all kinds of errors in measurement theory cannot be used in software, as there could be certain concepts in physical sciences pertaining to errors that cannot be applicable to software. In such a case, can there be some other classes of errors that exist in software measurements? All such possible errors
need to be listed and classified, as done in measurement theory, since measurement is a science of understanding errors as much as measurements according to Sydenham [SYDENHAM 1982].

5.2 Why standardization is missing in software?

Measurement (measurement of physical sciences) is in practice with a long history. It has been evaluated, tested, developed and used quite extensively. It is well developed and documented, and similar concepts are being followed in almost all countries. For example, for measuring mass- kilogram or pound are the units used. What can be said about software measurements in the same terms? Why do not we have standardization of software terms? It may be because software development started quite recently as compared to measurement theory. And in that small time, lots of terms and methods for performing various software tasks were discovered. Hence less time was devoted to the evaluation and analysis of these discoveries. We also have no laws for guiding development of software metrics as they are present in physical sciences. So today, almost every company that develops software decides its own measuring techniques/terms which the end user has to agree to. In such cases he may not be get what he ideally should have received. Whereas the same software developed by some other company may provide him with the desired deliverable. How can the end user save himself from such variations and receive a fair share? If we have standardization of all these terms and conditions the end user can ensure his fair treatment.
Standard specifications generally deal with standardization of terms and definitions, creation of standards of design/performance with the prescription of quality standards and the associated tests used to decide the quality, with rationalization of sizes and levels of quality to keep the options down to an efficient minimum number.

The way we have committees for preparation of standard specification for measurements in physical sciences, can we have the process followed for software too?

1. For creating standards in software, we can have a responsible, appropriate and authorized body consisting of people from different domains like software industries, research, educational institutes, etc along with the end user community, from different countries, to study the case and decide whether the specification is worthy. They can choose a drafting committee.

2. Then there could be a conference attended by relevant organizations that will eventually use those standards.

3. Upon agreement, the terms of reference can be decided and task be assigned to the drafting committee whose views would be taken to finalize the document and submit to an authoring body which would scrutinize the whole operation.

4. Upon satisfaction, or after a voting procedure the final draft can be issued as a declared standard.

5. This may first be considered at a national level and then extended to international level.

6. Creating international organization standards can require more stages and may be done by correspondence instead of face-to-face meetings.
7. These standard specifications need to be regularly reviewed so that they will remain an accurate statement of contemporary workable practice even as technology advances.

5.3 Software tools versus physical sciences tools of measurement

Properties that a hardware tool, used for measurement, must possess are accuracy, precision, traceability, sensitivity, error of measurement, ability to withstand reasonable shock loads, temperature excursions, relative humidity changes, ability to perform for long periods of operation and so on. Environmental factors and human behaviors have an ability to hamper an instrument’s performance.

How does this apply to software measurement tools?

Tools used in software measurement are software developed to ease-out the measurement operations. They measure certain qualities of software like complexity and effort. These tools are to be fed with certain inputs with reference to the code to be measured. The tools perform some calculations and provide output. These tools are operated by human beings and it is possible that human beings make errors while providing inputs which could be as miniscule as defining incorrect upper/lower input load capacities for the tool to function correctly. We yet do not have standardized tools that can parse through the code and take the required input on their own to perform measurement. Such tools remove the possibility of human errors completely. Only errors then, would be errors within the tools like mistake in the logic implemented. Software tools cannot be affected by human or environmental factors. Such tools are developed by
companies for their internal use. A tool developed by company A used to measure software developed by company B, may or may not be useful since all the required parameters for the tool may be required to be specified within the code and fed to the tool, which may not be what all companies do. We can have a software measuring tool comparable to the popular definition model of measurement in physical sciences. A software tape measure can be considered for the purpose of measuring lines of code. It can be a black box from which the tape can be pulled as the code develops. We can consider it to be a series of dots placed besides the code that will count the lines of code by comparing the code length with the dots, the way a measuring tape having numbers on it is used to measure the length of an object by placing the tape against the object.

So far, all measurements are performed after the code is developed, it should ideally be the case that measurements are performed before the coding stage and then the measurement values obtained after the developed system is measured should conform to the measurements taken before.

5.4 Work way up to concept formation from measurement method

It is necessary to have a proper definition of concept of the quality we want to measure before a method to measure it is developed. For example before coming up with a method to measure hardness of a metal we need to define what is hardness so that we know what we are trying to measure. Generally, to devise a measurement method, one starts from some direct concept of a quality and then seeks measurement scales. But in
some cases the concept of a quality arises from invariance in numerical laws arrived at by measurement, that is, it uses bottom-up strategy.

An example in measurement in physical sciences is Young’s modulus. This quality is arrived at from Hooke’s law: the observation that for an extensive class of materials, strain is proportional to stress.

In software we have Halstead’s metric Effort (E) [Halstead 1977] to consider for example. He assumes that any algorithm implementation consists of N selections from η elements. For selection of every element done with binary search, \( \log_2 \eta \) comparisons are required. A program is generated making \( N \times \log_2 \eta \) mental comparisons. Then he defines volume \( V = N \log_2 \eta \) and says ‘Volume is a count of mental comparisons and reciprocal of program level (1/L), which is a measure of average of number of elementary mental discriminations required for each mental comparison’, and comes up with Effort \( E = V/L \) and \( E = V^2/V^* \). He doesn’t consider the concept of effort before coming up with its measurement method.

We will take a look at a small class of software metrics called hierarchical software metrics to apply and compare properties and concepts with measurement theory. Hierarchical software metrics use graph theory, so the next chapters presents us with an overview of some basics concepts in graph theory and a summary of hierarchical software metrics.
CHAPTER 6
Graph Theory

6.1 Graph Theory Concepts

1. Graph: A graph or undirected graph $G$ is an ordered pair $G: = (V, E)$ that is subject to the following conditions:
   a) $V$ is a finite set, whose elements are called vertices or nodes,
   b) $E$ is a multiset of unordered pairs of vertices (not necessarily distinct), called edges or lines.

2. Degree:
   a) The degree (or valency) of a vertex is the number of edges incident (falling) on the vertex. The degree of a vertex $v$ is denoted $\text{deg}(v)$.
   b) The maximum degree of a graph $G$, denoted by $\Delta(G)$, is the maximum degree of its vertices.
   c) The minimum degree of a graph, denoted by $\delta(G)$, is the minimum degree of its vertices.

3. Path: A path in a graph is a sequence of vertices such that from each of its vertices there is an edge to the next vertex in the sequence. The first vertex is called the start vertex and the last vertex is called the end vertex.
4. Order: The order of a graph is $|V|$ (the number of vertices).

5. Digraph: A directed graph or digraph $G$ is an ordered pair $G = (V, A)$ with
   a) $V$ is a set, whose elements are called vertices or nodes,
   b) $A$ is a set of ordered pairs of vertices, called directed edges, arcs, or arrows.

   An arc $e = (x, y)$ is considered to be directed from $x$ to $y$; $y$ is called the head and $x$ is called the tail of the arc; $y$ is said to be a direct successor of $x$, and $x$ is said to be a direct predecessor of $y$. If a path leads from $x$ to $y$, then $y$ is said to be a successor of $x$, and $x$ is said to be a predecessor of $y$. The arc $(y, x)$ is called the arc $(x, y)$ inverted.

   In a directed graph, the indegree and outdegree of a node are the number of edges that arrive and leave the node respectively.

6. Complement of a graph: Complement of a graph $G$ is a graph $H$ on the same vertices such that two vertices of $H$ are adjacent if and only if they are not adjacent in $G$.
   a) To find the complement of a graph, you fill in all the missing edges, and remove all the edges that were already there.
   b) It is not the set complement of the graph; only the edges are complemented.
7. Connected Graph: A graph that has a path from any point to any other point in the graph is a connected graph.

8. Regular graph: A regular graph is a graph where each vertex has the same number of neighbors, i.e. every vertex has the same degree.

9. Doubly connected graph: A connected graph is doubly connected if its complement is also connected.

10. Complete graph: A complete graph is a simple (has no loops or multiple edges) graph in which every pair of distinct vertices is connected by an edge.
   a) The complete graph on \( n \) vertices has \( n \) vertices and \( n(n - 1) / 2 \) edges, and is denoted by \( K_n \). It is a regular graph of degree \( n - 1 \). Complete graphs are maximally connected.

11. Null Graph: The graph with no vertices and no edges is sometimes called the null graph or empty graph.
12. Flowgraph: A flowgraph is a directed graph in which two nodes, the start node, and the stop node obey special properties: the stop node has outdegree zero and every node lies on some walk from start node to the stop node [FENTON 1991].

6.2 Control Flow Structure

A lot of work in software metrics has been dedicated to the study of measures derived from the control flow structure of a program. This is modeled by directed graphs whose nodes correspond to program statements and where an edge from one node to another indicates a control flow between the corresponding statements. These directed graphs are normally called control flowgraphs or more simply flowgraphs [FENTON 1991].

Example:
10 INPUT P

20 Div = 2

30 Lim=INT(SQR(P))

40 Flag=P/Div-INT(P/Div)

IF Flag =0 OR Div=LIM THEN 80

60 Div=Div+1

70 Go TO 40

80 IF Flag<>0 OR P<4 THEN 110
90 PRINT Div;"smallest factor of";P;".

100 GO TO 120

110 PRINT P;"is prime."

120 END

Figure 6.6.1 A program and its corresponding flowgraph [FENTON 1991]

6.2.1 Flowgraph Model of Structure

Some commonly occurring flowgraphs: (Circles indicate start and stop nodes)
Figure 6.6.2 Some commonly occurring flowgraphs [FENTON 1991]
CHAPTER 7
Scales of Measurement

Data comes in various sizes and shapes and it is important to know about these so that the proper analysis can be used on the data. There are four scales of measurement that are usually considered.

The topic of types of scales used in measuring behavior can create a great deal of confusion in social and educational research. It is critical because it relates to the types of statistics you can use to analyze your data.

7.1 Types

[FENTON 1991], [FENTON 1997], [WIKIPEDIA 2008], [BECKER 1997], [BROWNE]

The class of admissible transformations for a given empirical relation system defines how unique each scale is and can also be used to define scale type.

Some important scale types:

1. Nominal
2. Ordinal
3. Interval
4. Ratio

Nominal
1. Nominal Data
a) The lowest measurement level you can use, from a statistical point of view, is a nominal scale.

b) A nominal scale, as the name implies, is simply naming or classifying of data into categories, without any order.

c) A physical example of a nominal scale is the terms we use for colors. The underlying spectrum is ordered but the names are nominal.

d) In research activities a YES/NO scale is nominal. It has no order and there is no distance between YES and NO.

And statistics

The most likely statistic that can be used with nominal scales is mode.

Ordinal

2. Ordinal Data

Ordered but amount of differences between values are not significant

a) e.g., political parties on left to right spectrum given labels 0, 1, 2

b) e.g., restaurant ratings

The simplest ordinal scale is a ranking. When a market researcher asks you to rank 5 types of beer from most flavorful to least flavorful, he/she is asking you to create an ordinal scale of preference.
There is no objective distance between any two points on your subjective scale. For you the top beer may be far superior to the second preferred beer, but to another respondent with the same top and second beer, the distance may be subjectively small.

And statistics

Ordinal data would use non-parametric statistics. These would include:

a) Median and mode
b) Rank order correlation
c) Non-parametric analysis of variance

Interval

3. Interval Data

Ordered, constant scale, but no natural zero

Differences make sense, but ratios do not (e.g., 30°-20°=20°-10°, but 20°/10° is not twice as hot!

a) E.g., temperature (C, F), dates

And statistics

Interval scale data would use parametric statistical techniques:

a) Mean and standard deviation
b) Regression
c) Analysis of variance
d) Factor analysis

e) Plus a whole range of advanced multivariate and modeling techniques.

Remember that you can use non-parametric techniques with interval and ratio data. But non-parametric techniques are less powerful than the parametric ones.

**Ratio**

4. Ratio Data

a) Ordered, constant scale, natural zero

b) E.g., height, weight, age, length

A ratio scale is the top level of measurement.

The factor which clearly defines a ratio scale is that it has a true zero point. The simplest example of a ratio scale is the measurement of length (disregarding any philosophical questions about defining how we can identify zero length).

And statistics

The same as for interval scales except they have true zero points.

A good example is the Kelvin scale of temperature. This scale has an absolute zero. Thus, a temperature of 300 Kelvin is twice as high as a temperature of 150 Kelvin.

When a scale consists not only of equidistant points but also has a meaningful zero point, then it is referred to as ratio scale. If we ask respondents their ages, the difference between any two years would
always be the same, and ‘zero’ signifies the absence of age or birth. Hence, a 100-year old person is indeed twice as old as a 50-year old one. Sales figures, quantities purchased and market share are all expressed on a ratio scale.

Ratio scales are the most sophisticated of scales, since it incorporates all the characteristics of nominal (definition of *nominal scale*), ordinal (definition of *ordinal scale*) and interval scales (definition of *interval scale*). As a result, a large number of descriptive calculations are applicable.
### 7.2 Summarizing the important scales

<table>
<thead>
<tr>
<th>Level</th>
<th>Can define…</th>
<th>Relation or Operation</th>
<th>Admissible Transformations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal</td>
<td>Mode, frequency</td>
<td>Equality (=)</td>
<td>(M' = F(M)) (F 1-1 mapping)</td>
</tr>
<tr>
<td>Ordinal</td>
<td>Median, percentile</td>
<td>Order (&lt;)</td>
<td>(M' = F(M)) F monotonic increasing i.e. (M(x) \geq M(y)) (\Rightarrow M'(x) \geq M'(y))</td>
</tr>
<tr>
<td>Interval</td>
<td>Mean, standard deviation</td>
<td>Subtraction (-) and weighted average</td>
<td>(M' = \alpha M + \beta) ((\alpha &gt; 0))</td>
</tr>
<tr>
<td>Ratio</td>
<td>Geometric mean, coefficient of variation</td>
<td>Addition (+) and multiplication ((\times))</td>
<td>(M' = \alpha M) ((\alpha &gt; 0))</td>
</tr>
</tbody>
</table>

Table 7.1 Some important scales [FENTON 1991], [FENTON 1997], [WIKIPEDIA 2008].

### 7.3 Scales with legitimate operations

[FENTON 1991], [FENTON 1997], [WIKIPEDIA 2008], [BECKER 1997]
Only certain operations can be performed on certain scales of measurement. The following list summarizes which operations are legitimate for each scale. Note that you can always apply operations from a 'lesser scale' to any particular data.

1. E.g. you may apply nominal, ordinal, or interval operations to an interval scaled datum.

Nominal Scale: You are only allowed to examine if a nominal scale datum is equal to some particular value or to count the number of occurrences of each value.

1. For example, gender is a nominal scale variable. You can examine if the gender of a person is F or to count the number of males in a sample.

Ordinal Scale: You are also allowed to examine if an ordinal scale datum is less than or greater than another value. Hence, you can 'rank' ordinal data, but you cannot 'quantify' differences between two ordinal values.

1. An example would be preference scores, e.g. ratings of eating establishments where 10=good, 1=poor, but the difference between an establishment with a 10 ranking and an 8 ranking can't be quantified.

Interval Scale: You are also allowed to quantify the difference between two interval scale values but there is no natural zero.
1. For example, temperature scales are interval data with 25°C warmer than 20°C and a 5°C difference has some physical meaning. Note that 0°C is arbitrary; so that it does not make sense to say that 20°C is twice as hot as 10°C.

Ratio Scale: You are also allowed to take ratios among ratio scaled variables.

1. Physical measurements of height, weight, and length are typically ratio variables. It is now meaningful to say that 10 m is twice as long as 5 m.

2. This ratio hold true regardless of which scale the object is being measured in (e.g. meters or yards). This is because there is a natural zero.

7.4 Frequently Asked Questions (FAQ)

What is a natural zero?

1. Some scales of measurement have a natural zero and some do not. For example, height, weight etc have a natural 0 at no height or no weight. Consequently, it makes sense to say that 2m is twice as large as 1m. Both of these variables are ratio scale.

2. On the other hand, year and temperature (C) do not have a natural zero. The year 0 is arbitrary and it is not sensible to say that the year 2000 is twice as old as the year 1000. Similarly, 0°C is arbitrary (why pick the freezing point of water?) and it again does not make sense to say that 20°C is twice as hot as 10°C. Both of these variables are interval scale.
CHAPTER 8
Design And Analysis of Hierarchical Software Metrics

8.1 Introduction
This chapter summarizes Prather’s hierarchical software complexity [PRATHER 1995].

There are two kinds of decompositions in the hierarchical structure:

1. Sequencing
2. Nesting

Each of the above extends recursively to the bottommost level consisting of irreducible flowgraphs.

8.2 The flowgraph Model

8.2.1 Flowgraph Definition:
A flowgraph $F$ is a directed graph consisting of

1. Process Nodes (outdegree = 1)
2. Decision Nodes (outdegree = 2)
3. Distinguished end (outdegree=0)
4. Distinguished begin (indegree = 0)
8.2.2 Cubic Graph

A cubic (undirected) graph G is a graph in which every vertex has degree 3. The following propositions are easily proven. Note that it is assumed that all graphs are connected.

**Propositions:**

1. Every cubic graph has an even number of vertices, $2n$
2. Every cubic graph G with $2n$ vertices has $3n$ edges.

8.2.3 Cubic flowgraph

If a cubic graph has $2n$ vertices, its order is said to be $n$, and author writes $o(F) = n$. Further he may consider half the vertices to be colored black and the other half white such that:

1. Black (decision) vertices have indegree = 1 and outdegree = 2
2. White (junction) vertices have indegree = 2 and outdegree = 1
3. Resulting directed graph is strongly connected
4. Insertion of finite number of (process) nodes of degree 2 (indegree = outdegree = 1) is permitted. These process nodes are inserted “on” or ”in” existing edges.
5. One edge is considered dotted (end to begin arc)

Removing the process nodes and ignoring the edge orientation and the vertex coloring, each cubic flowgraph is associated with a unique underlying cubic graph.
By eliminating the junction vertices, collapsing them as it were, while replacing the end-to-begin arc with a pair of distinguished begin and end nodes, we obtain a flowgraph.

### 8.2.4 Subflowgraph Notion

Given 2 flowgraphs $F$ and $F'$, $F'$ is considered to be a subflowgraph of $F$ if $F$ splits (Figure 8.1) along two oppositely directed edges, one of which may be dotted, such that their replacement by a single dotted edge on one side or the other yields $F'$. By virtue of having a proper subflowgraph, the flowgraph $F$ is no more than doubly connected, that is one can remove two edges (two oppositely directed splitting edges) so as to disconnect the graph.

![Diagram](image)

**Figure 8.8.1 Splitting of a flowgraph $F$ with respect to a proper subflowgraph $F'$**
Example 1.

Program name (list);

var

... 

begin

if A then

a

else

b;

α: c;

d;

while B do

begin

if C then

β: while D do

e

else

if E then

goto β

else

begin

f;
Figure 8.8.2(a) Pascal-like program text; (b) Flowgraph resulting from (a); (c) Corresponding cubic flowgraph with process nodes; (d) Cubic flowgraph after disregarding process nodes; (e) Underlying cubic (undirected) graph.
8.3 The Decomposition Theory

As seen in Figure 8.3, one has to identify an edge $v_1w_1$ and an edge $v_2w_2$ in the graphs $G_1$ and $G_2$, respectively to effect the replacements: $v_1v_2$ and $w_1w_2$ for the given pair resulting in the composite $G_1 \circ G_2$. Considering the oppositely directedness of edges we distinguish between a composition in sequence or nesting according to whether the first edge is dotted (second is assumed dotted in either case).

$v_1w_1$ and $w_2v_2$ are to be replaced by the directed edges $v_1v_2$ and $w_2w_1$ and in the case where $v_1w_1$ is dotted we take $w_2w_1$ to be dotted.
Using this technique Prather defines

1. Sequence (sum) $F_1 + F_2$ of the flowgraphs $F_1$ and $F_2$ and

2. Nesting $F(X_1) = F(X_1$ on $e_1)$ of $X_1$ in $F$ on the edge $e_1 = (v_1w_1)$, respectively.

Hence, Generalized Compositions:

1. $F_1 + F_2 + \ldots + F_s$ and

2. $F(X_1, X_2, \ldots, X_{(3n-1)})$

Where, $o(F) = n$ and $F_i, X_i$ are sub-flowgraphs of the composite.
8.3.1 Some conventions:

1. Null Flowgraph: $X_I = 0$
2. $F + 0 = 0 + F = F$
3. Elementary (process) flowgraph I’s with sequential iterates: $2I = I + I$, $3I = I + I + I$ and so on as in Figure 8.4.

The null flowgraph is just the dotted circular arc representing the “imaginary” arc from the end node to the beginning node. Since we can place process nodes “on” arcs, we get the equivalence shown in Figure 8.4.

![Figure 8.8.4 Degenerate cubic Flowgraphs used to represent (sequences of) elementary process(es).](image)
8.3.2 Proposition

1. \( o(I) = 0 \),

2. \( o(F_1 + F_2 + \ldots + F_s) = \Sigma_{i=1}^{s} o(F_i) \),

3. \( o(F(X_1, X_2, \ldots, X_r)) = o(F) + \Sigma_{i=1}^{r} o(X_i) \)

Example 2.

Composing the three flowgraphs of Figure 8.5 in sequence, a sum is obtained:

\[
F = F_1 + F_2 + F_3 = \text{Flowgraph F of Figure 8.2(c)}
\]

\[
F_1 = S (I, I) \text{ where S is the selection construct}
\]
Any flowgraph decomposes uniquely into sequentially irreducible sub-flowgraphs. They can be further be decomposed by nesting, that is, by factoring out maximal sub-flowgraphs. The flowgraphs on which they are nested are then irreducible both by sequencing and by nesting.
8.3.3 **Proposition: A flowgraph is prime if and only if it’s irreducible.**

Proof: If $F$ is not prime, then it splits as in Figure 8.1. If one of the edges exhibited there is dotted, then $F$ is sequentially reducible, and if not, then it is reducible by nesting, that is, $F$ is not irreducible. Conversely, if $F$ is reducible either by sequence or by nesting, then clearly it has a proper subflowgraph, so it is not prime.

**Corollary: A flowgraph is prime if and only if it is triply connected.**

An elementary process is modeled as a cubic flowgraph of order 0, having one process node and one dotted loop whose representative class is the dotted loop of Figure 8.6.
Theorem

(Sum of Primes Decomposition). Every flowgraph $F$ has a unique decomposition:

$F = P_1(X_{11}, X_{12}, \ldots, X_{1r_1})$

$+ P_2(X_{21}, X_{22}, \ldots, X_{2r_2})$

$+ \ldots + P_s(X_{s1}, X_{s2}, \ldots, X_{srs})$, 

Figure 8.8.6 Prime Flowgraphs of orders 0, 1 and 2.
where \( P_i \) are prime, \( r_i \leq 3o(P_i) - 1 \), and \( X_{ij} \) are non-null maximal sub-flowgraphs of \( P_i \).

**Corollary**

If \( F \) is a sequentially irreducible flowgraph of order \( n \), then the number of (non-null) maximal sub-flowgraphs of \( F \) is at most \( 3n-1 \).

**Example 3.**

For flowgraph of Fig. 8.2(c):

\[
F = F_1 + F_2 + F_3
= S(I,I) + R(I + I, G(X_1, X_2) + I) + I
\]

\( G \) is prime shown by VI in Figure 8.6.

The entire decomposition tree neglecting nested null subflowgraphs is as in Figure 8.7.
Figure 8.8.7 Decomposition tree for the cubic flowgraph of Figure 8.2(c)

Here it becomes necessary to distinguish the variants of selection and repetition constructs in regard to nesting of null or non-null subflowgraphs in the two positions left to fill. Disregarding the degenerate case where both positions are null six possibilities are seen owing to the selections construct’s symmetry. These are generally named D1, D2, D3, D4 and D5, the Dijkstra flowgraphs pictured in Figure 8.8. This naming has been superimposed on the decomposition tree of Figure 8.7.
8.4 Hierarchical Software Metrics

A flowgraph software metric \( m \) is a function

\[ m: \text{flowgraphs} \rightarrow \text{positive integers} \]

Metric \( m \) is hierarchical if:

Normalization: \( m(I) = 1 \),

Sequence: \( m(F_1 + F_2 + \ldots + F_s) = g_s(m_{F_1}, m_{F_2}, \ldots, m_{F_s}) \),

Nesting: \( m(F(X_1, X_2, \ldots, X_r)) = h_r(m_{X_1}, m_{X_2}, \ldots, m_{X_r}) \)

…inductively over the entire flowgraph decomposition structure.

8.4.1 Hierarchical definitions for a number of typical metrics:

\( \rho \): McCabe’s measure: \( E - N + 2 \)

1. \( \rho(I) = 1 \),

2. \( \rho(F_1 + F_2 + \ldots + F_s) = 1 + \sum_{i=1}^{s}(\rho F_i - 1) \), and
3. \[ \rho(F(X_1, X_2, \ldots, X_r)) = 1 + o(F) + \sum_{i=1}^{r}(\rho X_i - 1). \]

\[ \eta: \text{Number of (simple) paths metric} \]
1. \[ \eta(I) = 1, \]
2. \[ \eta(F_1 + F_2 + \ldots + F_s) = \prod_{i=1}^{s} \eta F_i, \text{ and} \]
3. \[ \eta(F(X_1, X_2, \ldots, X_r)) = \sum_{j=1}^{\eta F} \prod_{i=1}^{r} (\eta X_i)^{b_{ij}} \]

\[ \ldots \text{where } p_1, p_2, \ldots, p_{\eta F} \text{ is the set of simple paths of } F \text{ and } b_{ij} = 1 \text{ if } X_i \text{ is on path } p_j \text{ (else 0).} \]

\[ \mu: \text{Prather’s } \mu \text{ measure} \]
1. \[ \mu(I) = 1, \]
2. \[ \mu(F_1 + F_2 + \ldots + F_s) = 1 + \sum_{i=1}^{s} \mu F_i, \text{ and} \]
3. \[ \mu(F(X_1, X_2, \ldots, X_r)) = (1 + o(F)) \max \{ \mu X_1, \mu X_2, \ldots, \mu X_r \} \]

Illustrating the involved calculations considering \( \mu \) measure applied on flowgraph \( F \) of Figure 8.2(c) of example 3 and the decomposition tree of Figure 8.7:

\[ \mu(F) = \mu(D_2(I, I) + D_5(I + I, G(D_3(I), I + I) + I + I) + I) \]
\[ = \mu D_2(I, I) + D_5(I + I, G(D_3(I), I + I) + I + I) + \mu I \]
\[ = 2 \max \{ \mu I, \mu I \} + 2 \max \{ \mu(I + I), \mu(G(D_3(I), I + I) + I) \} + \mu I \]
\[ = 2 \max \{1, I\} + 2 \max \{1 + 1, \max \{2 \max \{1, 1 + 1\} + 1\} + 1 \}
\[ = 2 + 14 + 1 \]
\[ = 17. \]
Note for same $F$,

$F: \rho(F) = 6$ and $\eta(F) = 12$.

### 8.5 Prime flowgraph generation

Method of generating prime flowgraphs consists of the following outlined steps:

1. Color the vertices half black (decision nodes) and half white (junction nodes).
2. Establish an orientation of the edges consistent with the interpretation of vertex coloring.
3. Then identify a dotted (end-to-begin) edge.
4. At the end only prime flowgraphs are retained.

#### 8.5.1 Generation Method

1. Find all non-isomorphic triply connected cubic graphs on $2^n$ vertices.
2. For each isomorphism class, find all non-isomorphic 2-colorings of the vertices; retain only those having no chromatic circuits.
3. For each subclass, find all non-isomorphic orientations of the edges consistent with the vertex color requirement; retain only those that are strongly connected.
4. For each such pre-flowgraph, find all non-isomorphic ‘ways of identifying a single (dotted) edge.'
CHAPTER 9

Classical Measurements and Hierarchical Software Metrics Comparison

This chapter presents a comparison between hierarchical software metrics and measurement theory in physical sciences with respect to a philosophy of measurement, modeling and standardization. Hierarchical software metrics are used to measure program complexities.

9.1 Measurement Theory & Philosophy

9.1.1 Quality concept formation

Quality concept formation is extremely important in order to perform measurements since quality needs to be defined before developing measurement procedures and scales [SYDENHAM 1982]. Do we have a firm definition of the term complexity that we are trying to measure, for which we have so many metrics available? Some metrics devised for measuring complexity tend to measure the simple paths while some measure the depth of nesting, some measure the lines of code, some measure the number of independent paths, and some measure the elementary flowgraphs. We need to understand better what complexity is, if we intend to measure it. This definition needs to be accepted and used globally. We should be able to justify it to the potential user that this is how complex the software is. Thus, the formation of the concept for the quality we are trying to measure is essential. Without an appropriate definition of complexity we do
not know what is it that we are looking for, when we say the program complexity is so and so.

9.1.2 Uniqueness

Considering uniqueness, a measurement mapping $Q \rightarrow N$ does not determine a unique mapping. So we have admissible transformations by which one can convert a metric $M$ to another $M'$. Here $M$, $M'$ may be two different measurement representations for the same property, for example one may represent height in centimeters and other in inches. So height is the attribute that can be measured by using multiple metrics but all those have end results that can be inter-converted using an admissible transformation. This transformation must maintain the empirical relation. We don’t have any such transformations stated for software metrics.

9.1.3 Metrics Correlation

It is required that all metrics measuring the same attribute can be correlated in order to have the same end result by all methods which can be inter-converted using admissible transformations. We cannot correlate different hierarchical metrics say number of simple paths’ metric with Prather’s $\mu$ measure with number of paths measure in the way length in meters (in MKS system) can be correlated or inter-converted with centimeters (in CGS system) and vice-versa with a constant multiplier.
9.1.4 Consider equalities and inequalities mentioned by Prather in hierarchical metrics paper [PRATHER 1995]

He gives us a correlation between a few different hierarchical metrics, which gives us a ranking of one metric being bigger than the other and vice-versa, but doesn’t tell us the exact difference.

Consider for example,

1. \( \rho (F) = 1 + o (F) \) or

2. \( \rho (F) \leq \eta (F) \leq 2^{\rho (F)-1} \)

It would be good if we get a conversion factor on ratio scale instead of interval scale, i.e. instead of saying this metric falls within \( \geq \) something and \( \leq \) something it should tell us that it is a*metric where ‘a’ is some real number.

Let’s consider if the cyclomatic complexity result can be transformed to the complexity calculated by simple path metric, for instance. Where is the class of admissible transformations? What are the uniqueness conditions? We do not have answers for these questions.

Can we compare complexity of 2 programs?

Let P1 and P2 be two programs whose complexity is to be measured,

C1, C2 be two complexities respectively.

Let N1 and N2 be the end results of the measured complexities.

So if \( C1 \leq C2 \), is \( N1 \leq N2 \)?
It should follow this empirical relation. If this is fulfilled by one metric does it
apply to any other metric too? That is, is this empirical relation maintained using any
metric?

9.1.5 Units

We don’t have units like centimeters, seconds etc in software metrics as in
physical sciences, so we cannot find out the method used for measurement (like
McCabe’s method or Prather’s method or so on), merely looking at the end result. The
result tells us nothing about the metric used to perform measurement. What does it mean
when we say a program has complexity equal to 17, or 6, or 3, or 12, or 4, or 13? How
complex will the program be if its complexity is 100 or say 1000? Does it map to natural
numbers, real numbers or something else?

From the base units of SI system (Systeme International d’Unites) most other
units can be derived that are ever needed by multiplication or division of the base units
[SYDENHAM 1982]. Units need to be defined in software if we want to be able to use
software metrics on ratio scale that is if we want to be able to derive other units by
multiplication or division.

Presently software metrics can be considered to be using ordinal scale as we
cannot say how big is complexity (C1) as compared to complexity (C2) for programs P1
and P2 respectively if C1 = 12 and C2 = 9. We cannot use addition that is additivity
which is incorporated by ratio scale, when we only have ordinal metrics. So we need to
define what it means when complexity C is considered equal to some N that is we need to
have it defined as in how much is C = 9 or C = 12 or so on.
9.1.6 Questions to be asked to software metrics

1. What scale of measurement is used?
2. Does it have the formal properties of addition?
3. Can we consider this to be symbolic representation and measurement?
4. Should there be a maximum limit on the complexity of a program so that it can be called a good, testable and manageable program?
5. Can an individual having little knowledge about software complexity understand what does complexity of a program equal to 5 and of that equal to 15 mean? But an individual having little knowledge about length can understand clearly what would 1 meter and 5 meters be.
6. Does it follow the uniqueness conditions?

All these questions need to be answered for hierarchical software metrics to be considered as measurement as per the properties of measurements in physical sciences.

9.1.7 Measurement properties in Physical Sciences and Software

Summarizing classical measurement properties:

1. A measure is an objective description. Conversely it can be claimed that if we can arrive at a totally objective description of a property manifestation, the most vital step towards measurement has been taken.
2. Measures are descriptions of great conciseness. A single number tells us what it would take many words to express. Measurement gives a precise description pinpointing by a single number a particular entity, where a verbal description indicates a range of similar but differing things.
3. A measure of a property gives us an ability to express facts and conventions about it in the formal language of mathematics.

4. Measurement enables the measurand to be expressed in signals that can be handled by machines.

5. The representational theory should include an empirical relational system, a numerical relational system, a representation condition and a uniqueness condition.

6. For the measurement scale to be considered extensive it should have the formal properties of addition.

9.1.8 Hierarchical Software Metrics and Additivity

As per Prather’s axioms a hierarchical metrics satisfies the following three properties:

1. \( m(I) = 1 \)

2. \( m(F_1 + F_2 + \ldots + F_s) = g_S(mF_1, mF_2, \ldots, mF_s) \)

3. \( m(F(X_1, X_2, \ldots, X_r)) = h_F(mX_1, mX_2, \ldots, mX_r) \)

Prather [PRATHER 1995] says one can use any functions for \( g_S \) and \( h_F \) to introduce a new metric.

One can use functions like:

\[
 g_S(mF_1, mF_2, \ldots, mF_s) = \max\{mF_i\} \\
 h_F(F(X_1, X_2, \ldots, X_r)) = 2 \circ F \ast \min\{mX_i\}
\]
…but these are not additive. Also some of the metrics that Prather [PRATHER 1995] mentions do not follow additivity.

If we use a function like:

\[ g_s(mF_1, mF_2) = a(m(F_1) + m(F_2)) \]

\[ = a(m(F_1)) + a(m(F_2)) \]

\[ = g_s(mF_1) + g_s(mF_2) \]

Lets say we combine F1 and F2 using the operator ‘;’, then using the 3 axioms

\[ m(F_1; F_2) = m(F_1) + m(F_2) \]

That is,

\[ g_2(F_1 ; F_2) = g_1(mF_1) + g_1(mF_2) \]

\[ \Rightarrow mF_1 + mF_2 + 2 = mF_1 + 1 + mF_2 + 1 \]

Thus showing that these are additive.

Thus we can see that not all functions follow additivity and some may follow it. Perhaps additivity is not a property that must be demanded in software metrics. We can say at this point that hierarchical software metrics follow some properties of measurements and don’t follow some of them as can be seen in the following.

1. Some metrics are additive while some are not.
2. It can give us the number relational system as only positive numbers are considered in this case.
3. We have the mapping $M: \text{flowgraphs} \rightarrow \text{positive integers}$, so the representation condition is also fulfilled.

4. Uniqueness condition cannot be applied to these measurements as explained in the above sections.

This leaves us with a question. Is it necessary that all the properties of measurement in physical sciences have to be followed by software measurements? Is it only if all these properties are followed then software would be considered good software? It could be the case that not all properties of measurement theory are applicable to software; does that make software bad? It is necessary for software to have those properties by which the requirement specifications it is developed for are fulfilled.

9.2 Modeling

Models motor ingenuity i.e. they represent some aspect of real world and enable us to investigate the real situation without needing to actually produce it or modify an existing situation [SYDENHAM 1982].

There are different types of models like linguistic, iconic, mathematical, physical etc. In software science we generally use iconic type of models to represent the system that is to be developed. Hierarchical metrics can be considered to be using iconic and mathematical models, as it uses a tree structure decomposed till irreducible flowgraphs are reached and then calculates the end result.
9.2.1 System structure

Systems can be reticulated or box-cut, a process in which the whole is, for purpose of analysis, broken down into more detailed assemblies of subsystem blocks, the process being taken as far as is needed, perhaps finally to the basic elements [SYDENHAM 1982]. In software science, while developing software systems the Software Development Life Cycle is followed which can be considered as a process that is being broken down into more detailed assemblies of subsystem blocks called phases such as analysis, design, etc which further can be divided, perhaps, finally to basic elements, such as designing variables of a particular class, say, for the purpose of analysis. Hierarchical software metrics [PRATHER 1995] require the arrangement of underlying flowgraphs of a code in a tree structure using decomposition composed of sequencing and nesting extended recursively to a bottommost level consisting of irreducible flowgraphs those that cannot be constructed non-trivially from the sequencing and nesting operations.

Models must be proven and evaluated in some way to ensure that they do indeed represent the reality desired. Usually the models prepared first are proven inadequate in some respect and the process of refinement continues until one runs out of ideas, effort, or more optimistically, until the model provides an adequate level of simulation. Here comes the role of measurements in systems [SYDENHAM 1982]. Hierarchical software metrics map flowgraphs \( \Rightarrow \) positive integers.

The Popular Definition Model [SYDENHAM 1982] says measurement is comparing an unknown quantity against a defined standard for that kind of quantity. This
model indicates the need for an agreed standard unit for the parameter and a way to realize it. It also makes apparent, the need for a method by which the standard and the unknown are compared. We do not have any standards for calculating complexity so far defined in software that can be used to compare with the unknown quantity we want to measure. If we are trying to measure complexity with any of the methods given by Prather or McCabe or others, what is the standard scale of measurement we are comparing them to? So at present we cannot use comparison while calculating software complexity using hierarchical software metrics. But, there could be ways to do so like one mentioned in chapter 5 section 5.3 for calculating lines of code.

9.3 Standardization of measurement fundamentals and practices

According to Sydenham [SYDENHAM 1982] discrimination, repeatability, and accuracy are three essentials of measurement performance. All measurements are made by comparing the measurand against a defined standard in some way, like using direct method or indirect method etc. In software science we do not have any set standards (say some standard codes with assigned complexities) against which we can compare the calculated complexities of a piece of code in consideration. If we had a set of standard complexities, we could say we are using direct method of measurement, but since we need to calculate our results they cannot be directly obtained by comparison. Since we undergo calculations in finding our end-result in hierarchical metrics method, we can say we are using indirect measurements.
A measurand is compared against the defined standard, or something representing that standard, and the difference from the actual magnitude of the standard being expressed in subdivisions of the basic unit is used according to some form of scaling. Similarly in regards to complexity, can we have a basic unit that can be referred to for all kinds of codes that can be developed? In measurements in physical sciences the standard is a physical object or characteristic of a physical apparatus that represents the conceptual unit chosen to represent a particular measurable attribute. Though software is not a physical object like a piece of metal or land for instance, it is not as abstract as intelligence or beauty.

In classical measurements many standards exist such as national and international standards, a secondary standard arrived at, by some form of comparison with the primary standard or reference standard etc. From the reference standard other standards that can be used at the more hazardous working face called working standards or field standards are formed [SYDENHAM 1982]. In software we do not have any standards like international, national, global or secondary standards etc. As of yet, in software we do not have the concept of comparing a given piece of software with any of the existing pieces of code taken as reference for comparison and something that is universally accepted. But can this be done?

9.3.1 Standardization in software

Many authors are using the Dijkstra flowgraphs as basic prime flowgraphs in hierarchical software metrics. Just the way some authors refer to them, if all countries start using them it has the potential of becoming an international standard if agreed to,
universally. A thought can be given to the following way of using direct comparison method in software with respect to hierarchical software metrics. Prather uses a decomposition tree consisting of Dijkstra’s flowgraphs to calculate final results. We can create a toolbox called “Prime Flowgraphs Toolbox” by using different combinations of flowgraphs from those of Dijkstra’s flowgraphs and all possible prime flowgraphs that can be generated using the prime flowgraph generation method suggested by Prather. Such combination flowgraphs can be assigned unique names such as a, b, c, … or some other similar but unique symbols and their combinations can be assigned names in the following way: If flowgraphs ‘p’ and ‘q’ are nested on flowgraph ‘d’ then this combination can be marked as d (p, q) in the order of nesting. Such combinations can be performed in real time and need not be saved anywhere for future reference. Complexity measures or values can be assigned to these flowgraphs and hence to their combinations too in the toolbox. This method can be useful primarily for the following reasons:

1. To find the unknown complexity measure the trees can be compared directly with the flowgraph combinations from the toolbox. Such a method can be considered as a direct comparison method as in measurements in physical sciences.

2. Comparing complexities of two programs: Using direct comparison we can check if the decomposition tree of one is a subgraph of the other program’s decomposition tree as shown in the example flowgraphs of Figure 9.1 and thus find the required output. The figure shows that b (a) is
a subgraph of \( a(c, b(a)) \). This can help us further in finding complexities of the required flowgraph using direct comparison.

3. The similarity/dissimilarity between the subflowgraphs (and between their position in the combination flowgraph) of a ‘Prime Flowgraphs Toolbox’ can be related to the similarity/dissimilarity easily perceived in the numeric system. Considering an example of the number “2” appearing twice in the formation of another number – 2092, it makes immediate and logical sense that both the 2s in the number 2092 have different values of “2x1000” and “2x1” respectively. Extending this relation into the Prime Flowgraphs Toolbox method we may conclude that flowgraph “a” has a
different meaning in both the instances of its existence in the combination flowgraph ‘a (b, e, a)’.

Concept of the national measurement system (NMS) came into prominence in USA in 1960, which is a system of activity that can be given the credit of enabling manufacture, commerce, trade and communication to develop with some degree of compatibility between the different sectors of a nation’s economy and in international arrangements. This study arose due to three needs for quantifying data relating to effort expended on measurements to obtain a better understanding of

1. Basic measurements and their standards,
2. Data and standards on materials and
3. Information on technological standards and measurements [SYDENHAM 1982].

This could be considered in regards to software sciences too, so as to follow the same paths and procedures all over the world, which is not in practice at present.
CHAPTER 10

Conclusions

Thus, we saw how the concepts, properties and principles involved in software measurements relate to those in measurements in physical sciences. Also, the changes that need to be made in order for software measurement to be comparable with the classical measurement theory were seen. Differences in the two types of measurements with reference to properties, concepts, errors, units, standardization and measurement tools were presented too. A property that random errors are absent in software measurement was seen.

The class of hierarchical software metrics is used for this comparison. Using appropriate examples it was suggested how it would be beneficial to incorporate some practices of classical measurement theory into software measurement. Is it possible that software measurement can have some more properties altogether different from that of measurement theory of physical sciences? This thesis focuses on software measurement with respect to measurement in physical sciences. If one thinks of summarizing software measurement on its own, it may be possible to come up with some properties that software measurement has whereas measurement in physical sciences doesn’t.
REFERENCES


<http://web.uccs.edu/lbecker/SPSS/scalemeas.htm>

[BROWNE] BROWNE, H. “Ratio Scales”
<http://www.harrybrowne.org/articles/RatioScales.htm>


<http://en.wikipedia.org/wiki/Level_of_measurement>

<http://en.wikipedia.org/wiki/Young's_modulus>

<http://en.wikipedia.org/wiki/Hooke's_law>