Software Metrics & Software Metrology

Alain Abran

Chapter 7
Halstead's Metrics: Analysis of Their Designs

Agenda

This chapter covers:

- Introduction
- The definitions of Halstead's metrics
- An analysis of the design and definitions of 5 of Halstead's metrics, including:
  - entities and attributes measured,
  - scale types and
  - measurement units
  - easurement method
- A discussion on the findings

Agenda

This chapter covers:

- Introduction
- The definitions of Halstead's metrics
- An analysis of the design and definitions of 5 of Halstead's metrics, including:
  - entities and attributes measured,
  - scale types and
  - measurement units
  - Measurement method
- A discussion on the findings

Introduction

- Halstead's *metrics* are commonly known collectively as 'software science' [Halstead 1977]

  - Researchers have used them [Samoladas 2004] :
    - to evaluate student programs and query languages
    - to measure software written for a real-time switching system,
    - to measure functional programs,
    - to incorporate software measurements into a compiler, and more recently
    - to measure open source software

  - Halstead's metrics are included in a number of current commercial tools that count software lines of code.

Introduction

- A number of authors have adopted the structure of the Halstead's metrics as the basis for their own proposed measures:
  - Ex: the design of his Function Points method is based on the Halstead's 'volume' metrics [Albrecht 1984].

- In this chapter, we explore the various elements of the design of Halstead's metrics, including their definitions, objectives, scale types, measurement units, and measurement method
  - The term *metrics* is used in this chapter, rather than more precise terms like measurement method and measurement procedure from the metrology domain; readers will readily understand why we have done so in this chapter.

Halstead's Metrics: Definitions

- According to Halstead:
  - a computer program is an implementation of an algorithm considered to be a collection of tokens which can be classified as either operators or operands:
    - eg. a program can be thought of as a sequence of operators & their associated operands.
  - All Halstead's metrics are functions of the counts of these tokens:
    - By counting the tokens and determining which are operators and which are operands, the following base measures can be collected:
      - n1: Number of distinct operators.
      - n2: Number of distinct operands.
      - N1: Total number of occurrences of operators.
      - N2: Total number of occurrences of operands.
      - In addition to the above, Halstead defines the following:
      - n1*: Number of potential operators.
      - n2*: Number of potential operands.

Agenda

This chapter covers:

- Introduction
- The **definitions of Halstead's metrics**
- An analysis of the design and definitions of 5 of Halstead's metrics, including:
  - entities and attributes measured,
  - scale types and
  - measurement units
- A discussion on the findings

Halstead's Metrics: Definitions

- Halstead refers to n1* and n2* as the minimum possible number of operators and operands for a module and a program respectively.

  - This minimum number would be embodied in the programming language itself, in which the required operation would already exist (for example, in C language, any program must contain at least the definition of the function *main()*), possibly as a function or as a procedure:
    - n1*=2, since at least 2 operators must appear for any function or procedure:
      - 1 for the name of the function and
      - 1 to serve as an assignment or grouping symbol.
    - n2* represents the number of parameters, without repetition, which would need to be passed on to the function or the procedure [Menzies 2002].

Halstead’s Metrics: Definitions

- Halstead's metrics are all defined based on its set of base quantities (n1, n2, N1, N2, n1* and n2*).
  - The **length (N)** of a program P is:

$$N = N_1 + N_2. \qquad (1)$$

  - The **vocabulary (n)** of a program P is:

$$n = n_1 + n_2. \qquad (2)$$

  - The **volume (V)** of a program P is defined as:
    a) a suitable measure for the size of any implementation of any algorithm
    b) a count of the number of mental comparisons required to generate a program

$$V = N * \log_2 n. \qquad (3)$$

Halstead's Metrics: Definitions

- The next 5 of Halstead metrics are listed below

  (Details of the definitions and of their analysis are presented in the Advanced Readings section)

Box 1: Five of the Halstead Metrics

Program **potential (minimal) volume (V*)**:	$V^* = (2 + n_2^*) \log_2 (2 + n_2^*)$.	(4)
Program **level (L)**	$L = \frac{V^*}{V}$.	(5)
Program **difficulty (D)**	$D = \frac{1}{L}$.	(6)
The program **level estimator** ($\hat{L}$) of L	$\hat{L} = \frac{2}{n_1} * \frac{n_2}{N_2}$.	(7)
The **intelligent content (I)** of a program	P : $I = \hat{L} * V$.	(8)

See the corresponding analysis in the Advanced Readings section

Halstead's Metrics: Definitions

- Finally, the last 2 of Halstead's metrics are presented in detail in Equations 9 and 10.

  - **Programming effort (E)** is defined as a measurement of the mental activity required to reduce a preconceived algorithm to a program P. **E** is defined as the total number of elementary mental discriminations required to generate a program:
  - $$E = \frac{V}{L} = \frac{n_1 N_2 N \log_2 n}{2 n_2} . \qquad (9)$$

- In the definition of **E**, the unit of measurement of **E** is claimed by Halstead to be an *elementary mental discrimination*.

Halstead's Metrics: Definitions

- The required **programming time (T)** for a program P of effort E is defined as:

$$T = \frac{E}{S} = \frac{n_1 N_2 N \log_2 n}{2 n_2 S} \qquad (10)$$

  - where S is the Stroud number [1], defined as the number of elementary discriminations performed by the human brain per second.
    - The S value for software scientists is set to 18 [Hamer 1982].
  - The unit of measurement of T is the second.

[1] In 1967, psychologist John M. Stroud suggested that the human mind is capable of making a limited number of mental discrimination per second (Stroud Number), in the range of 5 to 20.

Agenda

This chapter covers:

- Introduction
- The definitions of Halstead's metrics
- An analysis of the design and definitions of 5 of Halstead's metrics, including:
  - **entities and attributes measured**,
  - scale types and
  - measurement units
  - measement method
- A discussion on the findings

Analysis of the Design of Halstead's Metrics

- ***The measurement objectives***
  - The objectives of most of Halstead's metrics are to measure the following attributes of a program:
    - length,
    - vocabulary,
    - volume (or potential volume),
    - level, difficulty, level estimator, and
    - intelligence content.
  - In addition, 2 of Halstead's metrics aim to measure something quite different, that is:
    - programming effort, and
    - required programming time.
  - These last 2 metrics refer to the measurement of entities of the development process, rather than to attributes of the source code.

Analysis of the Design of Halstead's Metrics

- **The measurand: the entities and the attributes measured**
  - **Entities**: the entities to which Halstead's metrics can be applied are the source code itself and the algorithm of that source code.
  - **Attributes**: All of Halstead metrics are based on identifying and counting only 2 types of attributes of the program to be measured:
    - the number of operators, and
    - the number of operands.
  - The empirical worlds of these 2 types of attributes can be easily mapped to a corresponding mathematical structure by respectively counting & adding the number of operators & operands in the source code (or the equivalent algorithm).

    - In the metrology sense, the number of operators and the number of operands correspond to '**base quantities**' in metrology.

Analysis of the Design of Halstead's Metrics

- However, to obtain a value for each of the 10 Halstead metrics, equations have to be computed.
  - Note that all the equations associated with the 10 metrics (Equations 1-10) correspond to a 'derived attribute' (or to a 'derived quantity', as defined by the *international vocabulary of basic and general terms in metrology* (VIM) – see side-box below on metrology terms and definitions).

Box 2: Metrology terms & definitions

Attribute: the property of an entity that can be determined quantitatively, that is, for which a magnitude can be assigned. In the metrology vocabulary, this is called a *measurable quantity*, or *quantity* for short.

Attributes of same type: attributes which can be placed in order of magnitude relative to one another.

Base attribute: a simple property defined by convention, with no reference to other attributes, and possibly used in a system of attributes to define other attributes; in this case, we talk about a *base attribute*. In the metrology vocabulary, this is called a *base quantity*.

Derived attribute: a property defined in a system of attributes as a function of base attributes. In the metrology vocabulary, it is called a *derived quantity*.

See also Appendix B: Measurement Definitions.

Agenda

This chapter covers:

- Introduction
- The definitions of Halstead's metrics
- An analysis of the design and definitions of 5 of Halstead's metrics, including:
  - entities and attributes measured,
  - **<u>scale types</u>**
  - measurement units
  - measurement method
- A discussion on the findings

Analysis of the Design of Halstead's Metrics

- **Analysis of the numerical world: scale types issues**
  - While Halstead's 10 equations do not appear overly complex, identification of their corresponding scale type is highly challenging.
    - For instance, it has been noted by Fenton [1997] that:
      - Equation 3 on program Volume is of the ratio scale type, while
      - Equation 5 on program Level is of the ordinal scale type.
    - By contrast, Zuse [1998] maintains that:
      - Equation 1 is of the ratio scale type, and
      - Equations 2, 3, 6, and 9 are of the ordinal scale type.
    - Moreover, it is not clear to which scale type Equations 4, 7, 8, and 10 belong.

Analysis of the Design of Halstead's Metrics

- This challenge in identifying the scale types in Halstead's metrics immediately points to major design issues, and, consequently, to practical issues:
  - for practitioners, when using and attempting to interpret the outcome of these metrics, as well as
  - for all other measures, the design of which has been derived totally (or partially) from Halstead's metrics such as in:
    - the initial design of Function Points and
    - subsequent measures based on the design of Function Points, such as:
      - Use Case Points,
      - Object Points,
      - etc.

Agenda

This chapter covers:

- Introduction
- The definitions of Halstead's metrics
- An analysis of the design and definitions of 5 of Halstead's metrics, including:
  - entities and attributes measured,
  - scale types and
  - measurement units
  - measurement method
- A discussion on the findings

Analysis of the Design of Halstead's Metrics

- ***Issues in measurement units***
  - The above statements on the scale types of Halstead's metrics also need to be revisited when the units of measurement in Halstead's equations are taken into consideration.

  For instance:

  - In Equation 1, the program ***length*** (N) is calculated by adding the total number of occurrences of operators to the total number of occurrences of operands.
    - However, since their units are different, operators and operands cannot be directly added unless the concept common to them (and its related unit) are taken into consideration, that is, 'occurrences of tokens':
      - then, the right-hand side of Equation 1 would give 'occurrences of tokens' as a measurement unit on the ratio scale:

$$N^{\text{occurrences of tokens}} = N_1^{\text{occurrences of operators}} + N_2^{\text{occurrences of operands}}$$

Analysis of the Design of Halstead's Metrics

- In Equation 2, the program ***vocabulary*** (n) is calculated by adding the number of distinct operators to the number of distinct operands:

$$n^{\text{distinct tokens}} = n_1^{\text{distinct operators}} + n_2^{\text{distinct operands}}$$

  - Again, since their units are different, distinct operators and distinct operands cannot be directly added, unless the concept common to them (and its related unit) are taken into consideration, that is, 'distinct tokens'.

  - This measurement unit must then also be assigned to the left-hand side of this equation, labeled 'vocabulary', and associated with the related concepts.

Analysis of the Design of Halstead's Metrics

- It can be noted that, while the concept of length is associated with a number to represent program size, the concept of vocabulary is not.

  - Indeed, the program *vocabulary* (n) reflects a different view of program size, being a measure of 'the repertoire of elements that a programmer must deal with to implement the program' (that is, the set of distinct tokens – n1 and n2).

  - Most probably, an expression such as 'size of a vocabulary' would have been more appropriate.

Analysis of the Design of Halstead's Metrics

- In Equation 3, program ***volume*** (V) has been interpreted with 2 different units of measurement:
  - 'the number of bits required to code the program' [Hamer 82] and
  - 'the number of mental comparisons needed to write the program' [Menzies 02] on the left-hand side of the equation:

$$V^{\text{bits or mental comparisons}} = N^{\text{occurrences of tokens}} * \log_2 n^{\text{distinct tokens}} .$$

- Thus, there is no relationship between:
  - the measurement unit on the left-hand side, and
  - the measurement units on the right-hand side of this equation.

Analysis of the Design of Halstead's Metrics

- Furthermore, on the right-hand side, the true meaning of the multiplication of 'occurrences of tokens' and 'distinct tokens' is not clear.
  - Such a multiplication would normally produce a number without a measurement unit – see side-box below.

Box 3: The measurement unit produced by $\log_2$

In general, in engineering applications, we do not take the logarithm of a dimensioned number, only that of a dimensionless quantity. For instance, in calculating decibels, we take the logarithm of a ratio of two quantities. A ratio of quantities with the same dimensions is itself dimensionless. We can write

$$log(a/b) = log(a) - log(b)$$

making it appear that we are taking the *logs* of dimensioned quantities *(a)* and *(b)*, but the dimensions 'come out in the wash': by the time we have finished (subtracting one *log* from the other), we have effectively taken the log of a dimensionless quantity *(a/b)*.

We can regard units as factors in an expression, for instance:

$$\begin{aligned} 8\ meters &= 8 * [1\ meter] \\ 800\ cm &= 800 * [1\ cm] \\ &= 800 * 0.01 * [1\ meter] \end{aligned}$$

In these terms, we have:

$$\begin{aligned} (8m)*\log_2(8m) &= 8*[1m]*\log_2(8*[1m]) \\ &= 8*[1m]*(\log_2(8)+\log_2[1m]) \\ &= (8*\log_2(8)+8*\log_2[1m])*[1m] \end{aligned}$$

The inconvenient term $8*\log_2[1m]$ is an additive term that depends on the units being used. If it is part of a valid engineering calculation, it will be canceled out somewhere in the process. It may be, for instance, that when we take the *log* of *8 meters*, we are actually taking the *log* of a ratio of *8 meters* to a *one-meter* standard length.

Analysis of the Design of Halstead's Metrics

- Equation 9 is used by Halstead to compute the ***effort*** (E) required to generate a program:

$$\overset{\text{elementary mental discriminations}}{E} = \frac{\overset{\text{distinct operators}}{n_1}\;\overset{\text{occurrences of operands}}{N_2}\;\overset{\text{occurrences of tokens}}{N}\;\log_2 \overset{\text{distinct tokens}}{n}}{\overset{\text{potential operators}}{2}\;\overset{\text{distinct operands}}{n_2}}.$$

B. The measurement unit on the left-hand side of this equation, referred to as 'effort', would be expected to be 'hours' or 'days'.

- Halstead, however, referred to 'the number of elementary mental discriminations' as the unit of measurement for the left-hand side.

C. Next, in the sense that the 'distinct operators', the 'distinct operands', and the 'occurrences of operands' are, in a generic sense, 'tokens', then it can be concluded the measurement unit on the right hand-side of this equation is a combination of these measurement units.

- Therefore, there is no relationship between the units of measurement on the left-hand and right-hand sides of Equation 9.

Analysis of the Design of Halstead’s Metrics

- Finally, Equation 10 is used to compute the required ***programming time*** (T) for the program:

$$T^{\text{seconds}} = \frac{n_1^{\text{distinct operators}}\; N_2^{\text{occurrence of operands}}\; N^{\text{occurrence of tokens}}\; \log_2 n^{\text{distinct tokens}}}{2^{\text{potential operators}}\; 18^{\text{psychological moments per second}}\; n_2^{\text{distinct operands}}}$$

  - Again, the measurement unit on the left-hand side, that is, seconds, does not in any way relate to the measurement unit on the right-hand side, which is, in fact, a combination of many different measurement units.
    - In the right-hand side of equation (10), Halstead refers to the ‘moments’ in this equation as “the time required by the human brain to perform the most elementary discrimination”, but no specific time measurement unit is specified.

Agenda

This chapter covers:

- Introduction
- The definitions of Halstead's metrics
- An analysis of the design and definitions of 5 of Halstead's metrics, including:
  - entities and attributes measured,
  - scale types and
  - measurement units
  - **measurement method**
- A discussion on the findings

Analysis of the Design of Halstead's Metrics

The Measurement Method

- All 10 of Halstead's metrics are based on the results of n_1, n_2, N_1, N_2, n_1*, and n_2*, which themselves must be based on a counting strategy to classify the program tokens as operators or operands.

- Unfortunately, there is a problem here in distinguishing between operators and operands.
  - This problem occurs because Halstead has provided an example with specific illustrations of operators and operands, but without generic definitions applicable to any program context. That is, Halstead
    - has not explicitly described the generic measurable concepts of operators and operands
    - and has asserted only that – in the example he provides – their description is intuitively obvious and requires no further explanation.
      - In practice, for measurement purposes, intuition is an insufficient means for obtaining accurate, repeatable, and reproducible measurement results.

Analysis of the Design of Halstead's Metrics

- Through his example, Halstead has not illustrated a 'measurement method', but merely a 'measurement procedure' – see side box 4 for further terms in metrology.
  - Therefore, in the specific context of Halstead's metrics, we use the less precise expression 'counting strategy' in this section, which, of course, may vary from one measurer to another.

Box 4: Further terms in metrology

Measurement method: a logical sequence of operations, described generically, used in the performance of measurements.

Measurement procedure: a set of operations, described specifically, used in the performance of particular measurements according to a given method.

See also Appendix B: Measurement Definitions.

Analysis of the Design of Halstead’s Metrics

- It is important, therefore, that the term ‘measurement method’ be clearly defined and consistently used, since all Halstead’s software science depends on counts of operators and operands [Lister 1982].
- However, more than 30 years later, there is still no general agreement among researchers on the most meaningful way to classify and count these tokens:
  - Individual researchers (and practitioners as well) must state their own interpretation or, alternatively, use one of the available counting strategies proposed by other researchers, such as in [Szentes 1986; Conte 1986 ; Abd Ghani 1996].
  - More recently, rule have been proposed for identifying operators and operands in the object-oriented programming (OOP) languages [Li 2004].

Analysis of the Design of Halstead's Metrics

- Of course, it is to be expected that different counting strategies will produce different values of n_1, n_2, N_1, and N_2, and, consequently, different values for the above ten equations.
- Furthermore, applying Halstead's metrics to either the source code or directly to the algorithm of the source code might produce different values for the same base quantities.

Box 5: The entity measured: the source code or the algorithm?

In Java language, the number of operators in the source code is different from the number of operators in the equivalent algorithm for that source code, since – as an example – in Java source code, each statement must end with a semicolon (;), which is an operator.

Agenda

This chapter covers:

- Introduction
- The definitions of Halstead's metrics
- An analysis of the design and definitions of 5 of Halstead's metrics, including:
  - entities and attributes measured,
  - scale types and
  - measurement units
- Summary: **A discussion on the findings**

Summary: Discussion on the findings

- In this chapter, Halstead's well-known set of metrics was analyzed by focusing on their design and, in particular, on their measurement units.

- The following comments can be made about them:
  - Based on the ISO International vocabulary of basic and general terms in metrology (VIM), Halstead's metrics consist of 10 derived quantities – Equations 1 to 10 – which use any of 6 base quantities (n_1, n_2, N_1, N_2, n_1^*, and n_2^*).

  - Halstead has not explicitly provided a clear and complete counting strategy to distinguish between the operators and the operands in a given program or algorithm.
    - This has led researchers to come up with different counting strategies and, correspondingly, with different measurement results for the same derived measures and for the same program or algorithm.

Summary: Discussion on the findings

- There is ambiguity and uncertainty about the scale types in Halstead’s metrics.

- There are major problems with the units of measurement for both the left-hand and the right-hand sides of most of Halstead’s equations.

- Implementation of the measurement functions of Halstead’s metrics has been interpreted in different ways than according to the goals specified by Halstead in their design.
  - For example, the program length (N) has been interpreted as a measure of program complexity, which is a different characteristic of a program [Fenton 1994].