Resolving the Mysteries of the Halstead Measures

Horst Zuse

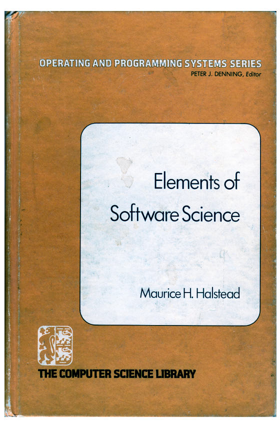
TU-Berlin, Franklinstraße 28/29
10587 Berlin
Germany
Phone: +49-30-881 59 88, Fax: +49-30-886 81 678
E-mail: horst.zuse@t-online.de
Internet: http://www.zuse.info

Abstract: The Halstead measures are still a subject of mystery. The properties of these measures are not well understood and there is still a confusion of the scales (types). Halstead's measures are used by many tools, but the usefulness and the meaning of the numbers of these measures are not clear. We use the concept of the extensive structure from measurement theory in order to investigate and resolve the mysteries of the Halstead measures.

Keywords: Maurice Halstead, measurement theory, software measurement, quality attributes, empirical, numerical, measure, complexity.

1 Introduction

The measures of the late Maurice Halstead were introduced in 1977 /HALS77/. These measure are widely used in software measurement tools and by many authors. For example, Oman et al. /OMAN90/ use these measures for the prediction of maintenance effort. Oman et al. reported about very strong correlations between the Measure Effort of Halstead and maintenance variables. However, they did not explain why. Halstead died in 1979 and for this reason he was not able to defend his measures against the upcoming criticism.



Picture 1: Maurice Halstead and the cover of the 1977 book /HALS77/.

The general problem with measurement values is the qualitative interpretation of the numbers. This is not only a problem of the scales (scales and scale types are different things), the major point is the type of the measurement structure behind the measures and its numbers. Halstead considers, among others, the software quality attributes length, volume, difficulty and effort. In physics, these quality attributes are clear defined. Length can be measured in cm and volume also is well defined. In daily life the term difficulty in any kind is connected with time and the term effort is expressed by imaginations of time. In measurement theory the extensive structure is one of the most important measurement structures. The qualitative attributes length, volume, time, money can be expressed by extensive structures. Qualitative attributes in general, but especially for software quality measurement, can be expressed by extensive structures, too. The concept of the extensive structure is a very powerful concept, because it clearly separates numerical from qualitative properties. Another thing is very important to mention: The basic idea of **measurement is**

the comparison of objects and numbers. Comparison is the important term! In measurement theory the definition of a measure does not consider units. The discussion of units is an important task, however, before we can do that the measurement structures and scales have to be determined.

The paper is structured as follows: In Chapter 2 the Halstead measures are introduced, in Chapter 3 measurement theory is explained, in Chapter 4 the Halstead measures are investigated and discussed, in Chapter 5 we give a summary of our results, Chapter 6 contains the conclusions and Chapter 7 presents the used literature.

2 Halstead's Measures

We introduce the five most important Halstead measures here. The measures of Halstead are based on the Quadruple (n1, n2, N1, N2), where n1 are the number of distinct operators, n2 are the number of distinct operands, N1 are the total number of used operators and N2 are the total number of used operands. Generally holds, that operands are variables and constants and operators are the other elements of programs. A clear definition of n1 and n2 is still missing, however, for our investigation this lack of a clear definition is without any relevance. Halstead's measures are defined as follows:

Measure N (Length of a program) : N = N1 + N2.

Measure n (Vocabulary): n = n1 + n2.

Measure V (Volume): $V = (N1 + N2) \log_2 (n1 + n2)$

$V = N \log_2 (n)$.

Measure D (Difficulty): $D = \frac{n1 * N2}{2 * n2}$

Measure E (Effort): $E = \frac{n1 * N2(N1 + N2) * LOG2(n1 + n2)}{2n2}$

E = D * V.

We now introduce the basics of measurement and the concept of the extensive structure.

3 Measurement Theory

Measurement theory considers, among others, the qualitative attributes of the considered objects and / or the qualitative (empirical) relations between the objects. Using measurement structures, different homomorphisms are defined in order to connect the qualitative attributes with the numerical representations. These homomorphisms also are called the representational theorem and lead, among others, to scales. We introduce here basics of measurement theory and the concept of the extensive structure.

Firstly, we introduce the empirical and numerical relational systems. Relational systems consist of a set of objects, relations between them and concatenation operations.

Empirical Relational System

Let **A** = (A, ≥, o) be an empirical relational system, where A is a non-empty set of objects, ≥ is an empirical relation on A and o is a closed binary operation on A (Of course, there are more than one relation and binary operation possible). According to Luce et al. /LUCE90/, p.270, we assume for an empirical relational system **A** *that there is a well-established empirical interpretation for the elements of A and for each relation ≥ of A*. We also assume the same for binary / concatenation operations. Binary and concatenation operations are used as synonyms.

Numerical Relational System

Let **B** = (ℜ, ≥, ⊗) be a numerical (formal) relational system, where ℜ are the real numbers, ≥ a relation on ℜ, and ⊗ a closed binary operation on ℜ. (Of course, there are more than one relation and binary operations possible). We also include the case that there are no relations or no operations. We call ⊗ a combination rule which is defined as u(a o b) = g (u(a), u(b)), where g is a function and u is a measure. If we have an additive combination rule, like u(a o b) = u(a) + u(b), then we can replace the sign ⊗ with + and we have **B** = (ℜ, ≥, +). Combination rules give important characteristics of software measures.

Definitions of Measures

We now introduce the basic definition of a measure. Firstly, we do not consider a concatenation operation and a combination rule. For this reason we have the both relational systems **A** = (A, ≥) and **B** = (ℜ, ≥) and a Measure u. We also write this as:

$$((A, \bullet\geq), (\mathfrak{R}, \geq), u).$$

The operations o and + have been left out because we only consider ranking structures. Then a measure based on ranking is defined as:

Definition 1: A **measure** is a mapping: u: A $\mathfrak{R}$ such that the following holds for all a,b $\in$ A:

$$a \geq b \Leftrightarrow u(a) \geq u(b).$$

Then the Triple (**A**, **B**, u) is called a **scale (not scale type!)**.
♦

The definition of a measure u above only considers ranking orders. It says that the empirical ranking order has to be preserved by the numerical ranking order or vice versa. It shows that the basic idea of **measurement is the comparison of objects and numbers**. A scale here is denoted as

$$(\mathbf{A}, \mathbf{B}, u) = ((A, \geq), (\mathfrak{R}, \geq), u).$$

A scale is a homomorphism between the two relational systems **A** and **B** by a Measure u. It is very important to mention, that scale types are not scales. Scale types are defined by admissible transformations. One more important aspect has to be mentioned: The definitions above do not deal with units. For this reason we will consider and discuss the Halstead measures without the discussion of units. The discussion of units is a subject after the determination of the measurement structures and scales. It is a subject of a further paper.

Additivity

However, we actually demand more of a measure. We want to have something above poor ranking or comparing of objects. We want to be additive in the sense that the combination of two objects is the sum of their measurement values. We want to consider the combination rule: u(a o b) = u (a) + u (b). Considering length of wooden boards that is a reasonable requirement. The sign o characterizes, for example, a concatenation of two wooden boards. In software measurement an additive property of the numerical relational system can be found very often /ZUSE98/. Additivity has an important impact on the relational systems. Formally, we need to speak of a binary operation o on the set A of objects - think of a o b as a combination of two objects (below, we will discuss this in detail). We want a real-valued function u on A that does not only satisfy

$$a \geq b \Leftrightarrow u(a) \geq u(b),$$

but also preserves the binary operation o, in sense that for all a,b $\in$ A

$$u(a \circ b) = u(a) + u(b).$$

We now introduce an extended definition of a measure which includes the additive property. Suppose we have the both relational Systems **A** = (A, ≥, o) and **B** = ($\mathfrak{R}$, ≥, +) and a Measure u. We write this as:

$$(\mathbf{A}, \mathbf{B}, u) = ((A, \bullet\geq, \circ), (\mathfrak{R}, \geq, +), u).$$

Then an additive measure is defined as:

Definition 2 (Measure based on an additive Homomorphism):

An **additive measure** is a mapping: u: A $\mathfrak{R}$ such that the following holds for all a,b $\in$ A:

$$a \geq b \Leftrightarrow u(a) \geq u(b),$$
and
$$u(a \circ b) = u(a) + u(b).$$

Then the Triple (**A**, **B**, u) is called a **scale (not scale type!)**..
♦

Both definitions of measures show that scales (not scale types) are not uniquely defined. It depends on the relational systems. According to this definition we see that measurement assumes, among others, a ranking and/or an additive homomorphism. The homomorphism describes rules for the mapping u: A $\mathfrak{R}$. The first rule says that the ranking properties have to be preserved, and the second rule - called an additive homomorphism - considers the additive operations of measurement values and the assigned empirical concatenation operation o. Again, the definitions of measures do not consider units. We consider units after the determination of the measurement structures and the scales. We now define the extensive structure /ZUSE98/.

Definition 3 (Extensive Structure)

Let A be a non-empty set, ≥ a binary relation on A, and o the closed binary operation o on A. The relational system (A, ≥, o) is a positive extensive structure if and only if (<=>) the following axioms hold for all a, b ∈ A:

A1: (A,≥) is a weak order

A2: a o (b o c) ≈ (a o b o c, axiom of weak associativity

A3: a ≥ b <=> a o c ≥ b o c <=> c o a •≥ c o b
axiom of monotonicity

A4: If c > d then for any a, b, then there exists a natural number n,
such that a o nc ≥ b o nd, Archimedean Axiom

♦

The extensive structure consists of empirical or qualitative conditions. The theorem of the extensive structure connects the extensive structure with an additive Measure u. In /ZUSE98/ we did introduce the theorem of the extensive structure.

Theorem (Extensive Structure)

Let A be a non empty set, ≥ is a binary relation on A, and o a closed binary operation on A. Then (A, ≥, o) is a closed extensive structure iff there exists a real-valued function u on A such that for all a,b∈ A:

$$a \geq b \iff u(a) \geq u(b)$$

and

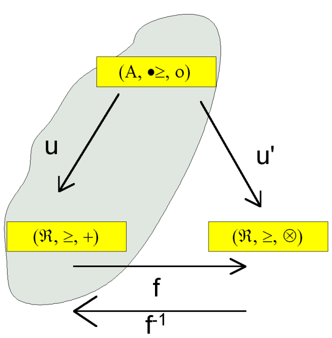
$$u(a \text{ o } b) = u(a) + u(b)$$

Another function u' satisfies the both statements iff there exists $\alpha>0$ such that

$$u'(a) = \alpha\, u(a).$$

♦

The next picture illustrates the connections of the empirical and numerical relational systems.



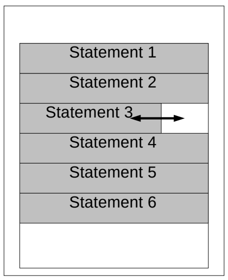
Picture 2: The mappings of the empirical and numerical relational systems by the Measures u and u’ and the functions f and f^{-1}.

The empirical relational system (A, ≥, o) consists of the considered objects A, the empirical relation ≥ between the objects (a ≥ b) and the (empirical) concatenation operation o between two objects (a o b) with a,bεA. The Measure u maps the empirical properties to the numerical ones with **B** = (ℜ, ≥, +) and the Measure u maps to the numerical relational system (ℜ, ≥, ⊗). The mapping from (A, ≥, o) to (ℜ, ≥, +) is described by the theorem of the extensive structure. Applying a strictly increasing monotonic function f to (ℜ, ≥, +) leads to the numerical relational system (ℜ, ≥, ⊗). It holds: u’ = f(u). The Measure u’ assumes the same extensive structure as the Measure u. However, the Measure u’ cannot be used as a ratio scale, anymore. The numerical transformation with f does not change the ranking order. The consequence is that non-additive measures assume an extensive structure, too. Since the function f is a strictly monotonic function the inverse function f^{-1} also is strictly monotonic function and it holds: $u=f^{-1}u'$. This shows, that non-additive measures which assume an extensive structure can be transformed to additive measures by f or f^{-1}. This concept is very important for the discussion of the Halstead measures.

We now consider the combination of two Measures u_1 and u_2 by an addition and a multiplication. For both Measures u_1 and u_2 we assume the extensive structure and an additive combination rule for a certain concatenation operation. Then for $u_+ = u_1 + u_2$ holds that the Measure u_+ also assumes an extensive structure. The Measure $u_* = u_1 * u_2$ does not assume an extensive structure in each case. We have to consider it in each case /ZUSE98/.

3 Investigation of Halstead Measures

We now consider the Halstead measures from a measurement theoretic view. We will discuss whether Halstead's measures assume an extensive structure or not. Using the extensive structure a concatenation operation based on the model of the Halstead measures has to be defined. The model of the Halstead measures is H= (n1, n2, N1, N2). A concatenation operation is defined as: g(a, b) = a o b: AxA   A, where A is a set of objects with a, b, aob $\in$ A. g is a function which concatenates two objects to a new one. The model for a program is: P = {S1, S2,..,Sn), where S1, S2, etc. are statements in the considered language. We introduce the following notation for programs: P_{si} = {Si) is a program with one statement Si. P_{arbi}= {S1, S2,…,Sn) is an arbitrary program and P_{large} = (S1, S2,..,Sn) is a large program. The definition of large programs is given below because it depends on the Halstead Measure n. We use the following concatenation operations: P = P_{arb} o P_s or P = P_{arb1} o P_{arb2} or P=P_{large} o P_{arb}.



Picture 3: A program P consists of statements Si.

We assume, that a program, module, class, etc. consists of a set of statements Si. For example, we have Statement S1. We can add Statement S2 writing it below S1. We can do this in an editor. Adding S2 below S1 is a concatenation operation P = P_{s1} o P_{s2}. Then, we can add Statement S3 to S1 o S2 and we get (S1 o S2) o S3. This is a concatenation operation, too. Adding statements are typical operations in the editor. We also can change the length of a statement Si. Let us consider the first two Statements S1 and S2. The we add Statement S3. However, we can add Statement S3 in different lengths and we always have a concatenation operation (S1 o S2) o S3. The following is important to say: Working with an editor implies automatically concatenation operations on the statement level Si. The described concatenation operations are always there, it is not necessary to define one artificially. It is no artificial definition of a concatenation operation, it is simply there.

Measures N and n

All the Halstead measures are based on the single Measures N and n. For this reason we consider these both measures very precisely.

Measure N

Firstly, we discuss the Measure N (Number of Operators and Operands): N = N1 + N2. We consider the concatenation operation of two arbitrary programs P = P_{arb1} o P_{arb2}. For the Measure N holds the following combination rule: N (P_{arb1} o P_{arb2}) = N1(P_{arb1}) + N2(P_{arb2}). It is easy to see that the Measure N assumes additivity which includes the assumption of the extensive structure via the theorem of the extensive structure. The consequence is that the Measure N can be used as an additive ratio scale. This result is consistent with our intuitive understanding of length. In physics, length also assumes an extensive structure and a measure for length, like a ruler, assumes an extensive structure and can be used as a ratio scale. In short: The Measure N has very nice and clear properties and can be used to analyze the length of programs based on the source code. The resulting numbers are clear (additive ratio scale). Assuming an extensive structure implies the condition of consistency (monotonicity). It means that changes in one or more statements causes consistent changes of the measurement of the whole module.

Measure n

With the Measure n = n1 + n2 the situation is a little bit more complicated. n1 is the number of operators and n2 is the number of operands. We have to discuss three different cases of concatenation operations and combination rules for the Measure n.

Case 1 (Measure n is additive):

We consider the concatenation operation P = (P_{s1} o P_{s2}). Since the programs P_{si} are very small programs (one statement) we have the following combination rule: n(P_{s1} o P_{s2}) = n1(P_{s1}) + n1(P_{s2}) + n2(P_{s1}) + n2(P_{s2}). The combination rule of the Measure n is additive. As mentioned above, this is the case with small programs. Adding a new statement P_{s1} adds usually new operands and new operators. In this case the Measure n assumes an extensive structure and can be used as a ratio scale. However, this situation is not a realistically one because programs are not always small, they grew from one version to another one.

Case 2 (Measure n violates the Archimedean axiom):
We consider the concatenation of two arbitrary programs: $P = P_{arb1}$ o P_{arb2}. The combination rule is: $n(P_{arb1}$ o $P_{arb2}) < n1(P_{arb1}) + n1(P_{arb2}) + n2(P_{arb1}) + n2(P_{arb2})$. We do not have an additive combination rule because the Archimedean axiom is violated. The reason is, that in P_{arb1} already operands and operators are defined which also occur in P_{arb2}. In this case the Measure n does not assume an extensive structure and cannot be used as a ratio scale. The measurement structures of the Measure n of the Cases 1 and 2 are differently. In Case 1 the Measure n assumes an extensive structure and in Case 2 it does not assume an extensive structure. It is important to mention: Obviously, the Measure n changes the structure of its qualitative properties. It becomes another measure and measures different qualitative attributes. The extensive structure describes the qualitative attributes of a measure, however, in Case 1 we have an extensive structure and in Case 2 we do not have an extensive structure.

Case 3 (Measure n becomes a Constant):
We consider the concatenation operation: $P = (P_{large}$ o Ps1). For the combination rule holds: $n(P_{large}$ o $P_{s1}) = n1(P_{large}) + n2(P_{large})$. The contribution of $n1(P_{s1})$ and $n2(P_{s1})$ is zero. The Measure n does not change its value. That happens because all the operands and operators of P_{s1} already are used in P_{large}. P_{large} is defined as a large program which contains all operators and operands of programs P_{si}. Of course, we have to investigate in reality when this happens.

Again, the Measure n changes its qualitative behavior or property from an extensive structure (Case 1) to the violation of the Archimedean axiom (Case 2) and finally to a simple constant (Case 3). This result has a very interesting consequence considering the other measures of Halstead. If you use the Measure n you cannot be sure whether it assumes an extensive structure or not.

Measure V
We now consider the Measure V (Volume) which is defined as: $V = N * \log_2 (n)$. The value of V is denoted as the volume of a program or *the implemented length of the algorithm*. One major problem with this Measure V is the name Volume of the measure. Volume is a well known characterization of the volume of objects, for example in mechanics or physics. Volume in physics is an extensive structure. Measures measuring the volume of objects in physics assume extensive structures and can be used as a ratio scales. In physics both lengths and volumes are extensive structures. Lengths and volume measures, like a ruler, assume extensive structures and can be used as (additive) ratio scales. In physics, volume has three dimensions, like: $V = a * b * c$.

The left part N of the Measure V is clear. N assumes an extensive structure, is additive and can be used as a ratio scale. The critical part of the Measure V is $\log_2 (n)$. Halstead explained $\log_2(n)$ as the necessary mental operations on the vocabulary n. Since we do not deal with units at this time, we drop off the units and the explanation of Halstead and consider this part of the Measure V from a measurement theoretic view based on the three discussed cases of the Measure n.

Case 1 (Measure n is additive and assumes an extensive structure):
Let us investigate $u = \log_2(n)$, where n assumes an extensive structure and can be used as a ratio scale. The Function $\log_2(n)$ is strictly monotonic increasing function. As we did show in Picture 2 it holds: Applying such a function to a measure does not change the ranking order and the consequence is that both Measure u and n assume the same extensive structure. The measure the same qualitative properties. For this reason we can drop off log2(n) and modify the Measure V to $V_{ext} = N * n$. This Measure assumes the same extensive structure as the Measure V. The numerical modification by dropping off $\log_2$ does not influence the extensive structure but it leads to the additive ratio scale. However, as shown above, the Measure n does not assume an extensive structure in each case.

Case 2 (Measure n violates the Archimedean axiom):
In this case the Measure n does not assume the extensive structure. This implies the violation of the extensive structure of the Measure V, too. It can be easily shown with an example. The consequence is that the Measures V changes from an extensive structure (Case 1) to a non-extensive structure (Case 2). The consequence is that the Measure V changes its qualitative properties which it measures.

Case 3 (Measure n becomes n = const):
In this case the Measure V changes to $V_{const} = N * \log_2(n=constant)$. We replace $\log_2(n=const)$ with k_v and get:

$$V_{const} = k_v * N.$$

The Measure V_{const} assumes an extensive structure and can be used as a ratio scale. The extensive structure of the Measure V_{cons}t is identically to the extensive structure of Measure N. This is a very interesting result. The Measure V converges to the Measure N for large programs. Both Measures V and N measure the same qualitative attributes for large programs. The constant k_v only is an admissible transformation of the ratio scale.

What are the consequences of this results? From our view the properties of the Measure V are changing from an extensive structure (Case 1) to a non-extensive structure (Case 2) and then to the identical extensive structure of Measure N (Case 3). The extensive structure of the Cases 1, and 3 are different extensive structures. Using the original Measure V it is not clear whether the resulting numbers assume the extensive structure or not. Compared to volume in physics it means, that volume assumes **sometimes** an extensive structure. This is an untenable state. In order to avoid unclear situations (Case 2) we recommend the following: Use the Measure N instead of Measure V because the Measure V converges to N for large programs. Doing this you have clear properties.

Measure D:

We now consider the Measure Difficulty (D), which is defined as:

$$D = \frac{n1 * N2}{2 * n2}$$

We modify the Measure D to: D = 0.5 * n1/n2 * N2. The Measure N2 assumes an extensive structure and can be used as a ratio scale. The interesting expression is n1 / n2. For large programs (see above) we can assume that n1/n2 becomes a constant k_d. Then the Measure D can be modified to

$$D' = k_d * N2.$$

This is very interesting result, again. The difficulty to write a program is based on the number of operands N2. The Measure D converges to Measure $D' = k_d * N2$ for large programs. Both Measures N2 and D' assume the same extensive structures and can be used as a ratio scale. It is a similar situation as with the Measure V. The Measure D changes its qualitative properties by assuming the extensive structure or not. Again, this is an untenable state.

Measure E:

The Measure Effort (E) is defined as : E = D * V. Paul Oman /OMAN90/ reports strong correlations between the Measure E and time based variables of maintenance effort. Other authors cannot get such correlations. Oman used very large programs. How can these different results be explained?

Considering the modifications above, the Measure E can be modified to: $E' = k_d\, N2 * k_v\, N$. We assume $k_e = k_d * k_v$ and we get

$$E' = k_e * N * N2.$$

We replace N2 = N – N1 and we get:

$$E = k_e * N\,(N - N1)$$
and finally
$$E' = k_e\, N^2\,(1 - N1 / N).$$

If we replace N with N = N1 + N2 then we get

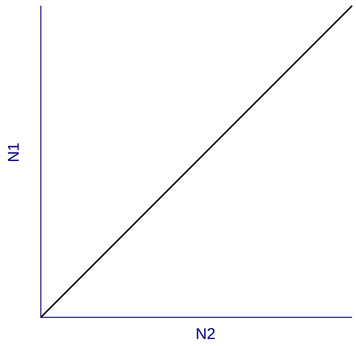
$$E = (N1/N2+1) * N2^2.$$

The interesting expressions are N1 / N or N1/N2. What are the values of this expression? Is it a constant for large programs? Does this expression change if we apply concatenation operation: $P = P_{large}$ o P_s or $P = P_{large}$ o P_{arb}?. Our hypotheses is that there is no or only a minimal change of the value of N1/N. Surely, it has to be investigated with experiments, but our hypothesis is that the expression becomes a constant for large programs P_{large}. Let us consider the expression N1 / N2. What does happen for large programs. We assume that N1 and N2 are linearily increasing. Again, the expression will be a constant or close to a constant..

We now assume that holds c = N1 / N = const. or C' = N1 / N2 = const. and we get for large programs

$$E' = q * N^2$$

with q = c + 1= const. This assumes that we have a correlation between N1 and N2. It should be investigated with real existing programs. We have the hypothesis that this is the case because the more operators and operands are existing the more effort is necessary.



Picture 4: Correlation between N1 and N2?

The Measure E' can be considered from two views. Firstly, it measures the effort to write a program. Then we have the empirical relation ≥ equally or more effort. Secondly, it can be considered as a prediction model predicting effort E'. In this case Effort is an external variable, for example money or person months. Firstly, we consider Case 1: The Measure N is an additive measure and assumes an extensive structure. As we did show in /ZUSE98/, but also showed in picture 2, the Measure N^2 assumes the same extensive structure as the Measure N and can be used as a non-additive ratio scale. The final result is that the Measure E' can be modified to E'' without changing the extensive structure. It holds:

$$E'' = q * N.$$

Both Measure E' and E'' assume the same extensive structures and can be used as ratio scales. Secondly, we consider $E' = q\, N^2$ as a prediction model. In a prediction model the variable E' is an external variable predicted by the internal variable or Measure µ. The question is whether

$$E' = q * N^2$$

is a proper prediction model. As we did prove in /ZUSE98/, Chapter 8, the following has to hold for a proper prediction function: If the predicted variable $V_{externa}$ assumes an extensive structure, like time or money, and it is used as a ratio scale, then the structure of a prediction function has to be:

$$V_{external} = a\, u^{b},$$

with a,b>0, where µ is an additive measure. The formula above is the only one possible structure of a prediction function if we assume the assumptions above. The variables a and b calibrate the values of $V_{external}$ in order to match the requirements of $V_{external}$.

5 Final Results

The Halstead measures for a long time were a subject of mystery and confusion. It was very difficult, almost impossible, to give an explanation of the meaning of the Halstead measures and their numbers. We now summarize our results.

The **Measure N** is clearly defined, it assumes an extensive structure and can be used as an additive ratio scale. The Measure N very often is denoted as the length of a program. Length is usually an extensive structure (in reality). Halstead also assumed that program length is an extensive structure.

We did show that the **Measure n** for small programs assumes an extensive structure, but for large programs we can assume n = const. This has important consequences for the Halstead measures below.

The **Measure V** is the most misunderstood Halstead measure. The qualitative property of the Measure V converges from an extensive structure to a non-extensive structure and then to the Measure N. The Measure V has unclear properties for small programs, for larger programs it converges to the Measure N. Our recommendation is to drop off the Measure V and to use the Measure N. Doing this you can avoid unclear properties for small and middle size programs.

The **Measure D** also is a subject of mystery. What is the meaning of the qualitative term difficulty? For large programs D converges to D', which is simply the Measure N2. Measure D' and N2 assume extensive structures and can be used as a ratio scale. In order to avoid problems use the Measure D'. Now, the term difficulty is clear, the more operands are used the more difficult is it to write a program.

The **Measure E** is a subject of mystery, too. What is the qualitative meaning of the term effort? We did resolve the mystery since the Measure E is based on the Measure N for large programs. Effort is the qualitative property of the Measure N. Effort can be seen as a prediction model. As we did show, it is a proper prediction model which predicts Effort by N.

Measure	**Case 1** Small programs	**Case 2** Middle size programs	**Case 3** Large programs with n = const.	**Comments**
Length: $N = N1 + N2$	Ext. structure, additive ratio scale.	Ext. structure, additive ratio scale	Ext. structure, additive ratio scale	Length N assumes an extensive structure and can be used as an additive ratio scale. **Proper Measure N**.
Vocabulary $n = n1 + n2$	Extensive Structure Add. Ratio Scale	No Ext. Structure Violation of the Archimedean axiom	n = const.	n changes the measurement structures.
Volume: $V = N \log_2 (n)$	$V_{ext}' = N * n$ Ext. structure. Identically (empirical) to $V = N \log_2 (n)$	$V = N \log_2 (n)$ No ext. structure, violation of the Archimedean axiom.	$V_{const} = k_v * N$ Extensive structure, additive ratio scale. Volume is Length.	V changes its qualitative properties. **Take:** $\mathbf{V_{const} = k_v * N}$.
Difficulty: $D = \frac{n1 * N2}{2 * n2}$	Unclear properties	Unclear properties	$D' = k_d * N2$ Extensive structure, additive ratio scale	D converges to D' for large programs / projects. Difficulty depends on N2. **Take** $\mathbf{D' = k_d * N2}$
Effort $E = D * V$	Unclear properties	Unclear properties	$E' = q * N^2$, $E'' = q * N$. Both measures assume the same extensive structure, non-additive ratio scale for E', additive ratio scale for E''. Prediction model: $E_{external} = a N^b$, a, b >0, a, b are calibration factors.	Effort converges to N for large programs. **Take E'' = q * N**. E'' assumes an extensive structure and can be used as a ratio scale. **Proper prediction model: E' = q * N²**

Table 1: Overview of Halstead's measures. Kv, kd and q are constants.

From our view the secrets and mysteries behind the Halstead measures are resolved. Length is a proper measure for program length (extensive structure and ratio scale). For large programs the quality attributes volume, difficulty and effort assume extensive structures and can be used as ratio scales. The only one measures which can be used properly are the Measures N and N2. The Measures V, D and E can be reduced or converged to the Measures N and N2: The Measure V converges to N, D converges to N2 and E converges to N. Of course, the assumptions should be investigated with experiments. However, from our view it makes sense to modify the Measures V, D und E as proposed above.

What is up with the units of the Halstead measures? Let us consider the original Measure N and the modified Measures V', D' and E'. The Measure N looks like the addition of two different values, namely operators and operands. The situation looks like adding apples and oranges. However, both are fruits and you can make a salad of it. You can say that you want to have two pounds of fruits consisting of one pound of apples and one pound of organges. Let us say that N1 and N2 are tokens of a program then N has the unit objects. We think in this way the question of units can be solved. The units of the modified Measures V' and D '. For the Measure and E' the unit is tokens.

6 Conclusions

On the first glance, the definitions and the properties of the Halstead measures are confusing. We did consider the famous measures of Halstead from a new measurement theoretic perspective. We did ask whether Halstead assumes for the software quality attributes, like length, volume, complexity, difficulty and effort extensive structures. The extensive structure is a very important measurement structure and leads to the ratio scale. The extensive structure describes the qualitative property of a measure.

We could show, that the original Measures Volume (V), Difficulty (D) and Effort (E) of Halstead can be reduced to the Measure N and N2 under certain conditions. For large programs Halstead's measures assume extensive structures for the Measures V, D and E. The Measure V converges to N, D converges to N2 and E converges to N. The presented results are surprising and we now hope that the use of the Halstead measures is more clear. It is an untenable situation if measures changes the properties depended on the structure and size of a program from an extensive to a non-extensive structure or from one to another different extensive structure.

In /OMAN90/ Oman et al. describe a strong correlation between maintenance variables and of the Halstead Measure Effort (E) for large programs. In the 90ties we had no theoretical explanation for this correlation, however, based on our investigation above such a correlation can be explained. The qualitative attribute Effort of the Halstead Measure E is the Length N of a program (for large programs). It assumes an extensive structure and can be used as ratio scales. There is the simple relationship: The longer a program the more effort. For small and middle large programs the Measure E does not assume an extensive structure, but the maintenance variables based on time assume an extensive structure. Correlations between such different properties and scale (types) are very critical.

Finally we state: From our view Maurice Halstead in 1977 did not know measurement theory. This is not a criticism. For this reason he was not aware of the extensive structure, their properties and the consequences of the scales, here the ratio scales. He could not know that it is difficulty to explain, that, for example, the Measure Volume changes its measurement structure and the scale (type) dependent on the structure and size of the programs (From ratio scale to ordinal scale and back to the ratio scale). The same holds for the Measures D and E. The basic ideas of the Measure N are ok. One more time, the powerful tools of measurement theory clear up the qualitative meaning of software measure.

We hope that our investigation helps to throw a new light on the matter of the work of Halstead. It also shall help to use his measures properly. One of the ideas of software measurement is to make better decisions.

7 Literature

In literature hundreds of papers can be found which consider the Halstead measures. For this reason we only refer to the original book of Halstead. Measurement theory related to software measurement in detail is considered in the book of the author.

/HALS77/ Halstead, M.H.: Elements of Software Science. New York, Elsevier North-Holland, 1977.

/LUCE90/ Luce, R. Duncan; Krantz, David H.; Suppes; Patrick; Tversky, Amos: Foundations of Measurement. Volume 3, Academic Press, 1990

/OMAN90/ Oman, Paul; Cook, C.: The Book Paradigm for Improved Software Maintenance. IEEE Software, Volume 7, No. 1, January 1990, pp. 39-45.

/ZUSE98/ Zuse, Horst: A Framework of Software Measurement. DeGruyter Publisher, Berlin, 700 pages, 1998.

About the Author

Horst Zuse was born on November 17, 1945. He received the Ph.D. degree in computer science from the Technische Universität of Berlin in 1985. Since 1975 he is a senior research scientist with the Technische Universität Berlin. His research interests are information retrieval systems, software engineering, software metrics and the measurement the quality of software during the software life-cycle. From 1987 to 1988 he was for one year with IBM Thomas J. Watson Research in Yorktown Heights. In 1991 he published the book: *Horst Zuse: Software Complexity - Measures and Methods* (De Gruyter Publisher). From August 1994 to December 1994 he was invited as a researcher by the *Gesellschaft für Mathematik und Datenverarbeitung (GMD)* in St. Augustin in Germany. In 1998 the book: *Horst Zuse: A Framework for Software Measurement* (DeGruyter Publisher) were published. From February 1998 to May 1998 he was a Visiting Professor with the University of Southwest Louisiana in Lafayette / Louisiana / USA. At December 1998 he received the habilitation (Privatdozent) in the area of *Praktische Informatik* (*Practical Computer Science*) by the department of computer science at the Technische Universität Berlin. Since November 2003 he is a visiting professor at the University for Applied Sciences in Senftenberg (FHL), too.