Empirical Software Engineering, 2, 313–316 (1997)

Comments to the Paper: Briand, Eman, Morasca: On the Application of Measurement Theory in Software Engineering

HORST ZUSE
Technische Universität Berlin, Berlin, Germany

The paper of Briand et al. criticizes the approach of using measurement theory in software engineering. A careful analysis of the statements of Briand et al. showed , that Briand et al. focus mainly on scale types. However, scales types are only one aspect of measurement theory, another important aspect of measurement theory are the empirical and numerical conditions that lead to hypotheses of reality. We now discuss some aspects of the paper of Briand et al.

Section 3.2 (Extensive Structure): Briand et al. Write on page 71: *Unfortunately, this misconception about the extensive structures has already had an important impact on the scientific work and the literature. Several authors have quoted Zuse's results in major journals or used his conclusions to validate their work.*

There does not exist a misconception of Zuse's work. I discuss the extensive structure, not because I believe that the axioms hold, but because they are important in order to understand the properties of measures. As a matter of fact many measures assume an extensive structure. If we have reasons to believe that some of the axioms do not hold, then this has consequences for software measurement. For example, if monotonicity does not hold then replacing in a program one component by a less complex one does not imply that the overall complexity decreases. This is an important problem. Also if monotonicity or commutativity do not hold, then all measures that assume this, have to be rejected, for example the Measure of McCabe and LOC. Hence, the prediction model of Basili $E = LOC^b$, which Briand et al. seem to accept, is not valid (it assumes an extensive structure, although Briand et al. criticize the extensive structure). From these examples we see that measurement theory helps to clarify the empirical assumptions of models and to avoid inconsistencies and contradictions.

Weyuker Properties: Briand et al. criticize my investigation of the Weyuker Properties. Firstly, it should be mentioned the following: Briand et al. write on page 70: *It is worth noting that in Zuse's description of this property (Zuse, 1991, p. 95), the existential quantifiers are substituted with universal quantifiers.*

That is not true. On page 1363 (Weyuker, 1988), we can find the statement:

$$(\forall P)(\forall Q)(|P| + |Q| \leq |P \ o \ Q|).$$

The statement is equally to

$$u(P \ o \ Q) \geq u(P) + (Q),$$

for all $P, Q \in \mathbf{P}$. **Here, Weyuker uses the all-quantifiers.** Also, Briand et al. write on page 70 (second paragraph): *These arguments are incorrect because properties W6 and W7 do not exclude the ratio scale, but only the ratio scale VIA the extensive structures (as mentioned . . .). Moreover, by applying the ratio scale admissible transformation, one can easily prove that properties W6, W7, W9 are meaningful for the ratio scale.*

Let us consider Property W7 in detail. It is the axiom of weak commutativity which is defined as: $u(P1\ o\ P2) = u(P2\ o\ P1)$ for all $P1, P2 \in \mathbf{P}$. Briand et al. write that this property is meaningful for the ratio scale. Yes, that is true. But, it is also meaningful for the ordinal scale (Roberts, 1979, p. 73), and even for the nominal scale. The question is what Briand et al. want to say with this statement? Showing that a statement is meaningful for a certain scale type does not imply that the measure can be used on this scale level. Every statement is meaningful for the absolute scale, but it can be easily shown, that not every measure can be used as an absolute scale.

Cost Estimation Models: In Section 7, page 82 Brand et al. consider cost estimation models. They give an example in order to show that scale transformations and the concept of meaningfulness are a hindrance for data analysis. They write among others: *According to the principles of measurement theory, there are admissible transformations applicable to each scale type. An examination of these transformations clearly indicates that non-linear transformations would not be admissible for interval or ratio scale levels. In the context of software engineering, this has some implications . . . They are meaningless because the transformation that are used are not admissible according to measurement theory.*
They consider the equation

$$E = aL^b, \tag{1}$$

with $a, b > 0$, E is effort and L some size measure. From (1) one derives

$$\ln E = \ln a + b \ln L, \tag{2}$$

which they claim is obtained by applying the non-linear logarithmic transformation on both E and LOC simultaneously. Because this is not an admissible transformation they conclude that in a measurement theoretic framework (2) cannot be derived from (1). I do **not** agree with their conclusions. From

$$E = aL^b,$$

we obtain

$$\ln E = \ln(aL^b). \tag{3}$$

That is **not** a scale transformation but a transformation of an equation. Measurement theory never forbids transformations of equations as long as they conform to the usual mathematical principles. In this case the logarithm may be applied because we can assume that E and aL^b are positive if E and L are ratio scales. Now, one can immediately see that (2) is a mathematical consequence of (3). If we would proceed as Briand et al. suggest, by transforming E and L simultaneously, we would get:

$$\ln E = a(\ln L)^b,$$

which has nothing to do with (2). Hence, the concept of admissible transformation is not a hindrance. On the contrary, it is useful in this case because it guarantees that E and L remain positive under all admissible transformations.

Science Is Not Mathematics: In Section 4, page 73 (second paragraph), Briand et al. also address the general problem of applying mathematics in empirical science. They quote: *Science is not mathematics*. First of all mathematics was introduced in software engineering long before measurement theory. Until now approximately one thousand software measures were defined. Each of these measures define by itself a mathematical model of complexity, maintainability, etc. So, why not use mathematics and especially measurement theory in order to study these measures and models? Doing this, we can see that there are several contradicting positions in this area. Here are two examples:

1. Many authors demand that the overall complexity can be obtained from the complexity of the components but this contradicts the axioms of Weyuker. Even Weyuker (1988, p. 1364), makes such a statement: *Although it seems reasonable that the complexity of a program body be related to the complexities of all its parts, it is difficult to determine the precise desired relationship.*

2. Many authors demand wholeness for a complexity measure but wholeness on effort does not imply this. Should this be ignored or should we try to find out what holds in reality?

I agree with Briand et al. that we do not yet know the scale type of software measures. And of course, we want to make experiments. However, also statistical methods assume empirically assumptions. They should be stated explicitly. For example, if we do regression analysis based on the Basili Model, we assume that we have an extensive structure and LOC is a ratio scale. This helps to interpret test results. For example, if we reject the extensive structure we might be more critical and search for better models. Hence, *Anything goes* in data analysis may not prevent us from reflecting what we are doing.

Final Statements: Briand et al. reduce their view of measurement theory to scale types. It is a widely spread misunderstanding that measurement theory is only good for scale types. Scale types are one thing of measurement theory. Another advantage of measurement theory are the empirical conditions combined with numerical conditions. They lead to hypotheses of reality. Of course, measurement theoretic assumptions should be combined with statistical assumptions and statistical techniques because the statistical results have to be interpreted empirically. Both disciplines are essential for software measurement and they are not contradicting, they even support each other.

References

Zuse, H. 1991. *Software Complexity: Measures and Methods*. Berlin, New York: DeGruyter.

Weyuker, E. J. 1988. Evaluating software complexity measures. *IEEE Transactions of Software Engineering* 14(9).

Roberts, F. S. 1979. Measurement theory with applications to decisionmaking, utility, and the social sciences. *Encyclopedia of Mathematics and its Applications*. Addison Wesley.

Horst Zuse received his Diploma Degree in electrical engineering from the Technische Universität in 1973, and 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 software engineering, software measurement, and quantitative methods of software engineering. From 1987 to 1988 he was for one year with IBM Thomas J. Watson Research in Yorktown Heights. In 1991 he published the book *Software Complexity—Measures and Methods, De Gruyter Publisher*. From 1989 till 1992 he was with the ESPRIT II Project 2384 METKIT (Metric-Educational-Toolkit) of the European Commission. Since 1990 he gives several times a year seminars about software measures for people of the industry within the scope of DECollege and ORACLE. He also gave many presentations and seminars about software measurement on conferences in US and Canada. Now, his research interests are validation of software measures, evaluation of cost estimation and prediction models, application of software measures in the whole software life-cycle, and standardization of software measures. From August 1, 1994 to December 31, 1994 he is invited as a researcher by the *Gesellschaft für Mathematik und Datenverarbeitung (GMD) St. Augustin (German National Research Center for Computer Science) (GMD)*. In October 1997 his book: *Horst Zuse: A Framework for Software Management*, will appear by the DeGruyter Publisher, Berlin, New York.