Response to: Comments on "Property-Based Software Engineering Measurement: Refining the Additivity Properties"

Lionel C. Briand, Sandro Morasca, *Member, IEEE Computer Society*, and Victor R. Basili, *Fellow, IEEE*

1 Introduction to the Response

POELS AND DEDENE have correctly identified a few inconsistencies related to the use of the union operator for modules of a modular system we defined in [1]. We gratefully acknowledge their comments, which show the increasing interest in the laying of theoretical bases for Software Engineering measurement.

We first show how the inconsistencies identified can be easily fixed without any conceptual change or addition to the axiomatic measurement framework proposed in [1]. Then, we discuss the more complex alternative proposed by Poels and Dedene.

The problems in [1] are listed below. Points 2 and 4 are actual inconsistencies, while points 3 and 5 are just redundancies. In what follows, we first report the original text of [1]; then we show how it should be corrected.

2 Explanation of Fig. 1—p. 70

Original Text

Union. The union of modules $m_i = <E_{mi}, R_{mi}>$ and $m_j = <E_{mj}, R_{mj}>$ (notation: $m_i \cup m_j$) is the module $<E_{mi} \cup E_{mj}, R_{mi} \cup R_{mj}>$. In Fig. 1, the union of modules m_1 and m_3 is module $m_{13} = <\{a, b, c, d, e, f, g, i, j, k, m\}, \{<b, a>, <b, f>, <c, d>, <c, g>, <d, f>, <e, g>, <f, i>, <f, k>, <g, m>, <i, j>, <k, j>\}$ (area filled with or or).

Modifications

Relationship <c, b> does not belong to the set of relationships of module m_{13}, the union of modules m_1 and m_3.

3 Property Cohesion.4—p. 77

Original Text

PROPERTY COHESION.4: **Cohesive Modules**. Let $MS' = <E, R, M'>$ and $MS'' = <E, R, M''>$ be two modular systems (with the same underlying system <E, R>) such that $M'' = M' - \{m'_1, m'_2\} \cup \{m''\}$, with $m'_1 \in M'$, $m'_2 \in M'$, $m'' \notin M'$, and $m'' = m'_1 \cup m'_2$. (The two modules m'_1 and m'_2 are replaced by the module m″, union of m'_1 and m'_2.) If no relationships exist between the elements belonging to m'_1 and m'_2, i.e., $InputR(m'_1) \cap OutputR(m'_2) = \emptyset$ **and** $InputR(m'_2) \cap OutputR(m'_1) = \emptyset$, then

$$[\max\{Cohesion(m'_1), Cohesion(m'_2)\} \geq Cohesion(m'') \mid Cohesion(MS') \geq Cohesion(MS'')] \quad (Cohesion.IV) \; \square$$

Modifications

The additional condition "If no relationships exist between the elements belonging to m'_1 and m'_2, i.e., $InputR(m'_1) \cap OutputR(m'_2) = \emptyset$ **and** $InputR(m'_2) \cap OutputR(m'_1) = \emptyset$" is redundant. It is already implied by the fact that $m'' = m'_1 \cup m'_2$.

4 Property Coupling.4—p. 78

Original Text

PROPERTY COUPLING.4: **Merging of Modules**. Let $MS' = <E', R', M'>$ and $MS'' = <E'', R'', M''>$ be two modular systems such that $E' = E''$, $R' = R''$, and $M'' = M' - \{m'_1, m'_2\} \cup \{m''\}$, where $m'_1 = <E_{m'1}, R_{m'1}>$, $m'_2 = <E_{m'2}, R_{m'2}>$, and $m'' = <E_{m''}, R_{m''}>$, with $m'_1 \in M'$, $m'_2 \in M'$, $m'' \notin M'$, and $E_{m''} = E_{m'1} \cup E_{m'2}$ and $R_{m''} = R_{m'1} \cup R_{m'2}$. (The two modules m'_1 and m'_2 are replaced by the module m″, whose elements and relationships are the union of those of m'_1 and m'_2.) Then

$$[Coupling(m'_1) + Coupling(m'_2) \geq Coupling(m'') \mid Coupling(MS') \geq Coupling(MS'')] \quad (Coupling.IV) \; \square$$

Modifications

The condition must be modified as follows.

Let $MS' = <E', R', M'>$ and $MS'' = <E'', R'', M''>$ be two modular systems such that $E' = E''$, $R' = R''$, and $M'' = M' - \{m'_1, m'_2\} \cup \{m''\}$, where $m'_1 = <E_{m'1}, R_{m'1}>$, $m'_2 = <E_{m'2}, R_{m'2}>$, and $m'' = <E_{m''}, R_{m''}>$, with $m'_1 \in M'$, $m'_2 \in M'$, $m'' \notin M'$, and $E_{m''} = E_{m'1} \cup E_{m'2}$ and $R_{m''} = R_{m'1} \cup R_{m'2} \cup \{<e_1, e_2> \in R \mid (e_1 \in E_{m1}$ *and* $e_2 \in E_{m2})$ *or* $(e_1 \in E_{m2}$ *and* $e_2 \in E_{m1})\}$. (The two modules m'_1 and m'_2 are replaced by the module m″, whose elements and relationships are the union of those of m'_1 and m'_2.)

If $m'' = m'_1 \cup m'_2$, then there would be no relationships in m″ connecting elements that originally were in m'_1 and m'_2.

5 Property Coupling.5—p. 79

Original Text

PROPERTY COUPLING.5. **Disjoint Module Additivity**. Let $MS' = <E, R, M'>$ and $MS'' = <E, R, M''>$ be two modular systems (with the same underlying system <E, R>) such that $M'' = M' - \{m'_1, m'_2\} \cup \{m''\}$, with $m'_1 \in M'$, $m'_2 \in M'$, $m'' \notin M'$, and $m'' = m'_1 \cup m'_2$. (The two modules m'_1 and m'_2 are replaced by the module m″, union of m'_1 and m'_2.) If no relationships exist between the elements belonging to m'_1 and m'_2, i.e., $InputR(m'_1) \cap OutputR(m'_2) = \emptyset$ **and** $InputR(m'_2) \cap OutputR(m'_1) = \emptyset$, then

$$[Coupling(m'_1) + Coupling(m'_2) = Coupling(m'') \mid Coupling(MS') = Coupling(MS'')] \quad (Coupling.V) \; \square$$

- *L.C. Briand is with the Fraunhofer Institute for Experimental Software Engineering (IESE), Sauerwiesen 6, D-67661 Kaiserslautern, Germany. E-mail: briand@iese.fhg.de.*
- *S. Morasca is with the Politecnico di Milano, Sede di Como, Piazzale Gerbetto 6, I-22100 Como, Italy. E-mail: morasca@elet.polimi.it.*
- *V.R. Basili is with the Institute for Advanced Computer Studies, Department of Computer Science, University of Maryland, College Park, MD 20742. E-mail: basili@cs.umd.edu.*

Manuscript received Feb 10, 1997; revised Mar. 18, 1997.
Recommended for acceptance by S.H. Zweben.
For information on obtaining reprints of this article, please send e-mail to: transse@computer.org, and reference IEEECS Log Number S96154.

Modifications

The additional condition "If no relationships exist between the elements belonging to m'_1 and m'_2, i.e., InputR(m'_1) $\cap$ OutputR(m'_2) = $\emptyset$ **and** InputR(m'_2) $\cap$ OutputR(m'_1) = $\emptyset$" is redundant. It is already implied by the fact that $m'' = m'_1 \cup m'_2$.

6 Discussion

As Poels and Dedene point out, it is important that inconsistencies be identified and removed. This will allow for a better understanding and refinement of the axiomatic framework proposed in [1]. In turn, it will lead to a more rigorous definition of software attributes and better measurement.

In their comments, Poels and Dedene have explored additivity, one of the most important and studied property in measurement. They propose the introduction of a new union operator for modules. They substantiate their idea by the fact that it is important to discriminate between modules that are disjoint and modules that, in addition to being disjoint, are not connected.

However, it is our position that we need to keep the set of operators as small as possible, since this will make it easier for researchers and practitioners to understand and discuss the properties proposed in [1] for different software attributes. That is why we introduced only a few operators (union, intersection, empty module, etc.) for modules, whose syntax and semantics were intentionally kept close to the syntax and semantics for sets, for which these operators are usually applied. One more union operator would be redundant, i.e., it would not add much expressiveness to the module "algebra" defined in [1]. If needed, the new union operator proposed by Poels and Dedene can be defined based on the other module operators and set operators.

Reference

[1] L.C. Briand, S. Morasca, and V.R. Basili, "Property-Based Software Engineering Measurement," *IEEE Trans. Software Eng.*, vol. 22, no. 1, pp. 68–86, Jan. 1996.