“Object-Oriented Products Metrics: A Structural Framework”

1. Concise Statement of Research Question

In spite of considerable prior research, a generic framework has not emerged for structuring work on object-oriented (OO) metrics. Developing such a framework for object-oriented product metrics is the focus of the current project. Thus the research question is:

"Can a framework be developed that captures the generic structure of the object-oriented products metrics space?"

2. Importance of the Research Question

A key research area in engineering of object-oriented software is measurement and the development of metrics [Henerson-Sellers 1996], [Lorentz and Kidd 1994]. A large number of object-oriented (OO) metrics have been proposed (e.g., [Henerson-Sellers 1996; Lorentz and Kidd 1994; Zuse and Drabe 2003] which have been summarized and studied (e.g., [Purao and Vaishnavi 2003], and classified into frameworks (e.g., [Abreau et al. 1994; Henderson-Sellers and Edwards 1993]). However, the dimensions of these frameworks are not well-defined, mixing the metaphors of process and product [Henderson-Sellers and Edwards 1993], or internal and external attributes (e.g. size, complexity, reuse, productivity, and quality [Abreu and Carapuca 1994]). In spite of considerable prior work, no generic frameworks are currently available to characterize the structure of the metrics space drawing on first principles such as mereology, set theory, and the theory of measurement. Such a framework could be useful in providing a deeper understanding of the existing metrics as well as that of potential metrics that do not currently exist.

3. Methodology

The proposed research will follow the design research methodology [Vaishnavi and Kuechler 2004] as sketched below:

**Awareness of Problem:** The awareness of the problem has been the result of a comprehensive survey of the literature [Purao and Vaishnavi 2003].
Suggestion: Object-oriented metrics can be classified into two related, but distinct, categories: product and process metrics [Purao and Vaishnavi 2003]. The former are concerned with properties of existing things, while the latter are concerned with the process of how the thing come to be. Product metrics, thus, deal with measurement of ‘things,’ treating the information system and its components as ‘things in the world’ in their own right. Following Bunge ([Bunge 1977], p. 17; [Wand and Weber 1990]), Information Systems are complex artifacts that contain component elements, each of which may contain its own properties. These component elements can be identified in terms of the underlying development paradigm. For example, in information systems built with the object-oriented paradigm, the fundamental building blocks may be objects, classified into classes, which may have attributes and methods, and the relationships among classes including abstraction and aggregation. Each element can possess its own properties, and may also have emergent properties, i.e. properties based on those of its
constituents. Theoretical approaches to model these properties must, therefore, combine a mereological and a set-theoretic perspective along with the relational theory of measurement.

**Development:** The Tentative Design is implemented in this phase. The techniques for implementation will of course vary depending on the artifact to be constructed. An algorithm may require construction of a formal proof. An expert system embodying novel assumptions about human cognition in an area of interest will require software development, probably using a high-level package or tool. The implementation itself can be very pedestrian and need not involve novelty beyond the state-of-practice for the given artifact; the novelty is primarily in the design, not the construction of the artifact.

**Evaluation:**

Validating the framework is a difficult proposition because it attempts to formalize and lend structure to the *theoretical* space underlying object-oriented metrics. Existing metrics provide only a partial manifestation of this underlying space. Clearly, the framework and accompanying axioms, constraints and lemmas must withstand the critical test of being able to represent existing research in object-oriented product metrics. This ability can provide *prima facie* evidence of the appropriateness of the proposed framework. Such an exercise can also result in identification of gaps and overlaps in current research.

To validate the proposed framework, it is necessary to show correspondence between the metrics predicted by the Framework and the underlying Metric-Space. The ability to represent metrics proposed by existing research in this space is, then, a necessary condition to demonstrate this correspondence. This necessary condition can be specified as:

\[ \text{Metrics-Space} \Rightarrow \text{Framework} \]

i.e. that there cannot be a metric in the metric space that is not implied by the framework. Verifying this would entail examining all existing metrics to ensure that the framework predicts them. A brute force

---

method to verifying this condition would require ensuring that the framework can explain all existing
metrics proposed by researchers. A corresponding, sufficient, condition needs to be verified to fully
validate the framework. This condition can be specified as:

\[ \text{Framework} \Rightarrow \text{Metrics-Space} \]

i.e. there cannot be a metric that is predicted by the framework but is not in the metric space. A brute
force method to verifying this condition would require generating all possible metrics based on the
framework, and ensuring that researchers have proposed or can propose them. This, however, is an
impossible task. The properties of artifacts are ill defined. Since each property is reified by one or more
metrics, there are countless ways of measuring the properties, and there are countless possible
variations of metrics. The theoretical metric space is, thus, vast, in stark contrast to the sparsely
populated manifested metric space. Our approach to test the sufficient condition, therefore, involves
examining summary outcomes from the repository and comparing these against expectations indicated
by the lemmas, followed by hypothesizing systematic extensions of the manifested metric space to
demonstrate that these variations either lead to other existing metrics or suggest potential metrics that
could exist in the theoretical metric space.

**Conclusion:** This phase is the finale of a specific research effort. Typically, it is the result of satisficing,
that is, though there are still deviations in the behavior of the artifact from the (multiply) revised
hypothetical predictions, the results are adjudged “good enough.” Not only are the results of the effort
consolidated and “written up” at this phase, but the knowledge gained in the effort is frequently
categorized as either “firm” - facts that have been learned and can be repeatably applied or behavior
that can be repeatably invoked - or as “loose ends” – anomalous behavior that defies explanation and
may well serve as the subject of further research.

4. **Expected Contribution**
The developed framework will have a number of implications for research. First, it will allow researchers to examine new metrics for their compliance with the framework. Since the framework will be built on a sound theoretical basis and validated with the existing metrics, conformance to the framework should add considerable weight to new metrics. Second, the different lemmas for the framework can help researchers and practitioners to automatically generate variations of existing metrics. Such generation can reduce the burden of creating new metrics, allowing researchers to analyze proposed metrics for their predictive potential [Fioravanti et al. 1999]. For example, any method-level metric may propagate in summative form at the class and system level.

Because product metrics play an important role in system development, the framework can also be useful for practitioners. By ensuring appropriate computation of measures of internal properties, the framework can allow practitioners to focus on the issue of appropriate linkages between internal properties and external characteristics such as testability, reliability and maintainability, which makes them very important to practitioners at all levels [El-Eman 2001; Meyer et al. 2001]. Such external characteristics are visible to the users, and therefore, remain the goal for measurement practices. They can, however, be obtained only when the information system is completed. Internal product metrics, on the other hand, cover properties visible to the development team and are thus obtainable during system development. They can therefore be used as leading indicators of the important external characteristics [Meyer 1998]. For example, if a relationship between a certain coupling metric and quality in terms of number of faults can be established, then one can reduce such faults and increase the product quality by minimizing coupling during design [El-Emam 2001]. Other examples for successful applications of OO metrics include prediction, management of cost and resources of developing systems [Fasolino et al. 2000], testing [Succi et al. 2003], and maintaining [Deligiannis et al. 2002; Sheldon et al. 2002]. Each of these uses can benefit from assurances of appropriate theoretical foundations for measurement of internal properties. The framework we have developed has addressed this goal.

References


