the points we raise, and it may well be that the information we seek has become a casualty of the pressure on space in academic journals. Nevertheless, it is vitally important to precisely specify the mapping from a language-independent set of metrics to specific programming languages and sets of observations. Such precision is particularly important since the source code used in published work is generally not publicly available. The usefulness of the proposed metrics (and others) will be limited until their application to specific languages is clearly specified. Failure to resolve such issues in the near future may impede the development and validation of effective OO software metrics. We believe that this is as important as the establishment of a sound theoretical basis for the metrics.

REFERENCES


Authors' Reply

Shyam Chidamber and Chris F. Kemerer

The main thesis of the Churcher and Shepperd comment is that it is important that the software metrics be clearly defined in order that other researchers can replicate the results, a point that we, of course, completely agree with. Therefore, we view this response to their comment not as a rebuttal, but merely as an opportunity to provide additional detail and insight to our thinking about how we developed the metrics for the readership of the IEEE TRANSACTIONS ON SOFTWARE ENGINEERING who may have interest in object-oriented metrics.

Churcher and Shepperd focus on the calculation of the number of methods per class, and suggest approximately half a dozen binary questions about what methods should either be included or not in the count. They then note that a potentially large number of different possible answers can be generated by combining all the possible combinations of answers to these questions.

Correction to "A Practical Approach to Programming with Assertions"

D. S. Rosenblum

In the above paper, a printing error resulted in the incorrect publishing of line 7 of Fig. 3 on p. 22.

The incorrect line read:

```
& & (int i=0; i < in size-1; i++) S[i] <= S[i+1] // S is ordered.
```

The correct line should read:

```
& & (int i=0; i < in size-1; i++) S[i] < S[i+1] // S is ordered.
```

Manuscript received February 6, 1995.
The author is with AT&T Bell Laboratories, Murray Hill, NJ 07974 USA (e-mail: dsr@research.att.com).
IEEE Log Number 9410122.