

Critical Research Issues for a “Science of Design”

UIUC–NSF Workshop on Continuous (Re)Design of Open-Source Software

Steve Schach
Vanderbilt University

It is essential to be able to measure design quality. Without appropriate design metrics, it is impossible to quantify design. Consequently, a Science of Design requires an associated set of metrics for measuring design quality; without such metrics, we have no way of determining whether future designs produced using a Science of Design are better or worse than current designs.

My primary interest in open-source software is in the maintainability of open-source software. I measure maintainability by measuring common coupling, a design metric. The *coupling* between two units of a software product is a measure of the degree of interaction between those units and, hence, of the dependency between the units. Coupling has been shown to be connected to fault-proneness. Coupling has not yet been explicitly shown to be related to maintainability. On the other hand, there is as yet no precise definition of maintainability, and therefore there are no generally accepted metrics for maintainability. Nevertheless, if a module is fault-prone then it will have to undergo repeated maintenance, and the resulting frequent changes are likely to compromise its maintainability. Furthermore, these frequent changes will not always be restricted to the fault-prone module itself; it is not uncommon to have to modify more than one module to fix a single fault. Consequently, the fault-proneness of one module can adversely affect the maintainability of a number of other modules. In other words, it is easy to believe that common coupling can have a deleterious effect on maintainability.

Three of our research projects relate to the quality of the design of open-source software.

1. Stephen R. Schach, Bo Jin, David R. Wright, Gillian Z. Heller, and A. Jefferson Offutt, “Maintainability of the Linux Kernel,” *IEE Proceedings--Software* **149** (February 2002), pp. 18-23. (A preprint is available at www.vuse.vanderbilt.edu/~srs/test/linux.first.pdf)

We downloaded 365 versions of Linux. For each version in turn, we examined the 17 kernel modules and counted the number of lines of code in each module. Then we counted the number of instances of common (global) coupling between each of the kernel modules and all the other modules in that version of Linux. We obtained two primary results:

First, we found a linear dependency between lines of code and version number that is significant at the 99.99% level. Second, we found that the number of instances of common coupling grows exponentially with version number.

We concluded that, unless Linux is redesigned with little or no common coupling, it seems inevitable that, at some future date, the dependencies between modules induced by common coupling will render Linux extremely hard to maintain.

2. Ligu Yu, Stephen R. Schach, Kai Chen, and Jeff Offutt, "Categorization of Common Coupling and its Application to the Maintainability of the Linux Kernel," submitted. (A preprint is available at www.vuse.vanderbilt.edu/~srs/test/linux.second.pdf)

We applied a new common coupling categorization, based on definition-use analysis, to version 2.4.20 of Linux for Intel-based computers. We counted the number of instances of common coupling between each of the kernel modules and all the other modules, both kernel and nonkernel. We found that the version of Linux we examined contains 99 global variables. We determined whether each of the 15,710 instances of those global variables is a definition or a use of that global variable. We found 2,013 unsafe definitions of global variables, that is, definitions of global variables that could affect a kernel module if a modification involving that global variable were made to the module in which the global variable is defined.

Currently we are investigating earlier versions of Linux, in order to measure the extent to which the quality of the design has decreased with successive versions.

3. Ligu Yu, Stephen R. Schach, Kai Chen, and Jeff Offutt, "A Comparison of Common Coupling within Linux and FreeBSD," in preparation.

It has been suggested that the high levels of common coupling in Linux are unavoidable because extensive common coupling is essential for all operating systems. To determine whether this is the case, we are currently investigating a broad variety of other operating systems. Our examination of FreeBSD shows that FreeBSD has only 69 global variables and only 1,019 instances of those global variables. Most important, we found only 82 unsafe definitions of global variables. Thus, on the basis of our metrics, FreeBSD appears to be far better designed than Linux.

Linux is a poster child for the open-source movement, but its design quality is low. A Science of Design will hopefully lead to better quality software. However, the only way we will be able to determine whether a future Science of Design is indeed an improved paradigm is to develop appropriate design metrics. Without such metrics to test our outcomes, there seems little point in developing a Science of Design.