



A RESEARCH TOOL FOR SOFTWARE DESIGN AND DOCUMENTATION

David A. Scanlan
MIS Department
California State University
Sacramento, California 95819

Currently there are no good research tools for studying the effectiveness of software design and documentation techniques. Usually the research tools used are paper-and-pencil measures where the subjects design a program or attempt to comprehend a previously written program using one of two or more design techniques. With paper-and-pencil measurements, the researcher usually counts the number of syntax and logic errors the subject makes. Also, the researcher may give multiple-choice questions which are designed to measure comprehension. Using paper-and-pencil measurement tools, the researcher only obtains information on the finished product; therefore, most of the programmer's behaviors exhibited during the writing of or comprehension of a program are not measured.

I have written a ram-resident program (tool) which will record all programmer activity. The program records the programmer's activity while s/he is in an editor writing and correcting the program, and it records the program's behavior while compiling and executing. Thus, the tool records all keystrokes and combinations of keystrokes and the exact moment in time when they occurred. It records all run-time and compile-time errors and the time when they occurred. The execution of this tool, as well as the data files, are transparent to the user. The tool is written for Turbo Pascal. Versions for other compilers and interpreters are now being developed.

The ability to "capture" the above data gives software engineering researchers one of the most powerful tools in existence. This tool will allow researchers to monitor programmer's behavior while in a laboratory or in the field. Combined with "paper-and-pencil" measurement tools, just about every meaningful programmer behavior can be measured.

Since very little is known about the effectiveness of design and documentation techniques, this tool will make it very easy to compare the effectiveness of Nassi-Shneiderman, Warnier-Orr charts, decision trees, decision tables, finite-state diagrams, HOS charts, action diagrams, flowcharts, pseudocode, structured English, and others.

Almost any area in software psychology could benefit from this tool. For example, researchers could measure the effectiveness of:

1. levels of program indentation
2. graphical vs. non-graphical computer languages
3. debugging aids
4. comments
5. using an editor while programming
6. pretty print
7. modularity
8. various types of error messages
9. program syntax
10. various icons
11. etc., etc.

Since this tool can monitor a student's behavior while s/he is writing programs, the tool can be used to measure the effectiveness of various teaching methods and materials.

This tool is licensed for research purposes only. Only a nominal copy and service charge will be required.