Currently Available Program Testing Tools: Technology and Experience

Leon G. Stucki
McDonnell Douglas Corporation

Complementing the 1975 International Conference on Reliable Software, held April 21-23 in Los Angeles (see May Computer), a workshop was held April 24-25 to survey the current status of program testing tools. Co-sponsored by the U.S. National Bureau of Standards, ACM, and the IEEE Computer Society (Technical Committee on Software Engineering), the workshop was held at the Airport Marina in Los Angeles on 24-25 April 1975. Attendance was 160 and represented many diverse interests. The fact that software is rapidly becoming recognized, by those outside as well as inside the programming profession, as one of the most complex human activities yet undertaken—set the stage for the discussions which emerged.

Three types of activities were provided at the workshop: selected presentations, a panel discussion, and working groups. The presentations addressed four major arenas for test tool development. A sampling of applicable technologies was provided in each of the following areas:

- Testing Strategies for Large-Scale Systems
- Testing Techniques for Operating Systems
- Testing Tools for General Users
- Testing Tools in the Hands of Researchers

A panel discussion on “Currently Needed Program Testing Tools” presented views representative of a broad set of interests. And finally, a series of birds-of-a-feather sessions allowed time for exploration of special interest topics.

Presentation Highlights

The session on “Testing Strategies for Large-Scale Systems” began with a talk on “The Need to Salvage Test Tool Technology” given by Marj Kirchoff of McDonnell Douglas, who pointed out our tendency to discard many of the successful techniques we have developed for specialized applications when the sponsoring project has been completed. Development of a test tool specification library was suggested for consideration as a promising approach to the preservation of the technology. Other talks in this session by S.A. Steele of RCA, M.A. Ikezawa of Logicon, and J. Rizza/D. Hacker of TRW conveyed various tools and technologies which have been applied to large scale systems.

The session on “Testing Tools for Operating Systems” discussed several novel approaches now being applied to operating systems. The paper by M.D. Hopwood/J. Lockett described the RAND Monitor/Stimulus-Generator. This hardware tool is used to introduce a load on a computing system and measure the resultant response time. Pre-stored messages are used to stimulate the system with their respective responses times being recorded. Joseph Buechler of Boole and Babbage discussed the use of a sampling technique for performance measurement and evaluation. The application of the sampling approach in a number of commercially available tools was described. Tom Bredt of Stanford emphasized the need to develop systematic methods for analyzing operating systems if our goal of reliable operating systems is to be realized. A class of analysis methods applicable to both hardware and software systems was discussed. Mike Burlakoff described an integrated set of software tools being designed at Wright Patterson AFB to reduce the...
design, development, and verification efforts for avionic software.

A talk on the National Software Works (NSW) Project by Steve Crocker and Bob Balzer of USC-ISI highlighted the afternoon session on “Testing Tools for General Users.” This new venture undertaken by ARPA is designed to test the concept for using the ARPA Network for providing access to remotely located programming tools. The National Software Works is being designed to provide centralized facilities for (1) log-in, tool invocation, and help procedures; (2) parameter and file specification; (3) access control of both tools and user files; (4) accounting for usage-based charging; (5) automatic movement of files to and from the site where a tool is; and (6) a catalog of what tools are available.

A special Birds-of-a-Feather session was held in the evening. This session explored in detail many of the practical issues involved including distribution, control, and security aspects of the NSW.

A sampling of program testing tool builders then provided a glimpse into a number of tools that are currently available for a variety of scientific and commercial environments. Those tools described included McDonnell Douglas’ Computer Program Management Technique (CPMT) approach to automating software development utilizing FORTRAN tools such as PET and JOYCE for automatic program analysis and documentation; General Research’s RXVP system for FORTRAN program analysis; Caine-Farber-Gorden’s Structuring Engine for automatically generating structured programs from existing unstructured FORTRAN code; CAPEX’s COTUNE II, an execution frequency analyzer for COBOL programs; Synergetics’ PRO/TEST Library of Testing Software for COBOL programs; and Applied Data Research’s flow charters and program construction aids for COBOL.

The session entitled “Testing Tools in the Hands of Researchers” offered a look at some areas currently being investigated at various universities. M.A. Hennell, University of Liverpool, described a facility designed to examine some of the problems which arise in the implementation of high quality numerical software libraries.

W. E. Howden of U.C. San Diego discussed systems for automating the generation of program test data and the problems associated with this very complex problem. R. E. Fairley of Texas A&M and H. A. Schmid of the University of Toronto both described experimental interactive systems for the development and testing of Algol 60 type languages. V. R. Basili and M. V. Zilkowitz of the University of Maryland described checkout compilers which they have built for Algol and PL/I type languages. The advantages of placing additional testing features directly into future compilers was strongly pointed out.

### Panel Discussion Highlights

On the second day a panel discussion was chaired by John Brown of TRW on the topic of “Currently Needed Program Testing Tools.” Panelists included Carlo Broglio (FAA), Dave Oppenheim (Abacus Programming), Rona Stillman (U.S. National Bureau of Standards), and Leon Stucki (McDonnell Douglas). The ensuing discussion provided an interesting melange for the theoreticians, tool builders, and user communities. Several concrete suggestions came out of the discussion including the need to begin establishing standards for the testing of software, the need to encourage manufacturers to start building hooks into future compilers which will allow program structural data and symbol table information to be readily accessible to testing functions, and the need to pass on the knowledge and experience of a few large companies to a much broader programming community. A hierarchy of testing activities was suggested for panel discussion and drew a heavy response from the audience. Considerable audience concern was expressed for particular user problem areas. One example offered was the need to know that critical software used to design and simulate nuclear processes has been adequately tested (e.g., will not cause or lead to a future nuclear power plant disaster).

The large attendance and active participation of those who did attend pointed out the timeliness and complexity of the topic covered. The knowledge shared and dialog begun should prove very beneficial for future software development.