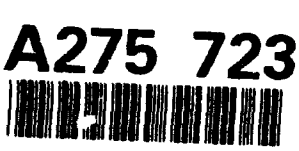


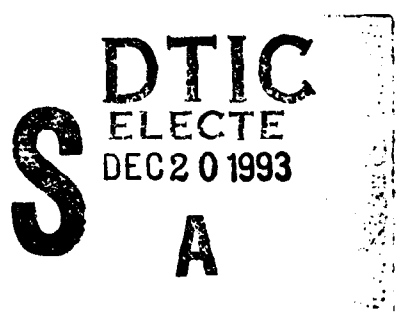
(12)

Final Report

ONR Contract N00014-85-K-0414
Theoretical and Pragmatic Issues in Software Engineering

1 April 1985 – 30 September 1990

Martin Davis and Elaine Weyuker



257
250

Although software engineering evolved pragmatically from efforts to develop practical paradigms for the development of reliable large-scale software systems, our work was based on the belief that significant progress would result from the kind of fundamental understanding that could only be obtained from deeper understanding of the underlying theoretical issues using models of computation fundamentally different from those studied in the now classical theory of computation.

Our research has included relatively abstract theoretical work, empirical studies, and system implementation. Our underlying belief was that software engineering research should follow the standards typical of most fields of science and engineering, in which researchers develop theories based on accepted models in the field and practical observations which are then validated by empirical studies. We have seen our work as an integrated whole in which experiments and implementations can suggest theoretical developments which in turn lead to further empirical work.

Beginning at the theoretical end of the spectrum, continuing previous investigations of formal notions of criteria for adequacy of software test data, we studied such a notion based on measures of *distance* bewteen programs satisfying the axioms for a metric space. We found analogues of the concept of *critical point* emerging naturally from our work. We were also able to prove a new version of our earlier *dichotomy theorem*. In addition, we investigated the formalization of what properties are desirable for any software test data adequacy criterion. These properties were then used to compare several well-known testing criteria, and strengths and weaknesses were noted. In a similar vein, properties of good software complexity measures were also proposed, and an assessment of well-known complexity measures performed. Again, the strengths and weaknesses were noted.

Substantial progress was made in our ongoing work on data flow testing. In particular, a prototype tool, ASSET, was designed and implemented, and experimentation began to assess the usefulness of these proposed techniques. New theoretical work was begun to consider the effect of unexecutable program segments on the use and applicability of data flow testing. Interesting and surprising results showed that unexecutable paths can substantially change the relative effectiveness of various well-known testing strategies. A large empirical assessment of the cost of using data flow testing was performed during this period. We found that even the most demanding of our data flow strategies was surprisingly inexpensive to use, when assessment was based on the number of test cases needed to satisfy the selected data flow criterion. We also found, however, that the number of unexecuable definition use associations could be quite large, and that it can be costly and time consuming to make this determination. Finally, we developed algorithms to use data flow information as the basis for selecting regression test suites.

In a totally different vein, we also designed and implemented a specification-based

testing tool to help manage both the development of test cases, and their use in regression testing for large software systems.

Other work included a new domain-based notion of software reliability that had the unique property that it incorporated a notion of criticality of failure into the measure.

A different area of research included an analytic comparison of partition testing strategies and random testing. Our conclusions were somewhat different from those of earlier work done by others that used simulations to make the comparisons.

PUBLICATIONS

Selecting Software Test Data Using Data Flow Information, *IEEE Trans. on Software Engineering*, Vol.SE-11, Apr 1985, pp. 367-375. (S. Rapps and Elaine Weyuker)

ASSET: A System to Select and Evaluate Tests, *Proceedings of the Conference on Software Tools*, New York, N.Y. April 1985, pp. 72-79. (P. Frankl, S. Weiss and Elaine Weyuker)

A Data Flow Testing Tool, *Proceedings Softfair II*, San Francisco, Ca., Dec 1985, pp. 46-53. (P. Frankl and Elaine Weyuker)

Design for a Tool to Manage Specification-Based Testing, *Proceedings of the Workshop on Software Testing*, Banff, Canada, July 1986, pp. 41-50. (T. Ostrand, R. Sigal and Elaine Weyuker)

A Generalized Domain-Based Definition of Software Reliability, *Proceedings of the Workshop on Software Testing*, Banff, Canada, July 1986, pp. 98-107. (S. Weiss and Elaine Weyuker)

Data Flow Testing in the Presence of Unexecutable Paths, *Proceedings of the Workshop on Software Testing*, Banff, Canada, July 1986, pp. 4-13. (P. Frankl and Elaine Weyuker)

Axiomatizing Software Test Data Adequacy, *IEEE Trans. on Software Engineering*, Vol. SE-12, No. 12, Dec 1986, pp. 1128-1138. (Elaine Weyuker)

An Overview of Software Testing Research, *Proceedings of the Fourth International Conference on Testing Computer Software*, Washington, D.C., June 1987, pp. 11-28. (Elaine Weyuker)

How To Decide When To Stop Testing, *Proceedings of the Fifth Annual Pacific Northwest Software Quality Conference*, Portland, Oregon, Oct. 1987, pp. 145-154. (Elaine Weyuker)

Metric Space-based Test-data Adequacy Criteria, *The Computer Journal*, Vol. 31, No. 1, 1988, pp. 17-24. (Martin Davis and Elaine Weyuker)

Influences of Mathematical Logic on Computer Science, IN *The Universal Turing Machine - A Half-Century Survey,* Rolf Herken, editor, pp. 315-326. Verlag Kemmerer & Unverzagt, Hamburg, Berlin 1988; Oxford University Press, 1988. (Martin Davis)

The Evaluation of Program-Based Software Test Data Adequacy Criteria, *Communications of the ACM*, June 1988, pp. 668-675. (Elaine Weyuker)

An Empirical Study of the Complexity of Data Flow Testing, *Proceedings of the Second Workshop on Software Testing, Verification, and Analysis*, Banff, Canada, July 1988, pp. 188-195. (Elaine Weyuker)

Evaluating Software Complexity Measures, *IEEE Trans. on Software Engineering*, Vol. 14, No. 9, Sept 1988, pp. 1357-1365. (Elaine Weyuker)

Using Data Flow Analysis for Regression Testing, *Proceedings of the Sixth Annual Pacific Northwest Software Quality Conference*, Portland, Oregon, September 1988, pp. 233-247. (T. Ostrand and Elaine Weyuker)

An Applicable Family of Data Flow Testing Criteria, *IEEE Trans. on Software Engineering*, Vol. 14, No. 10, Oct 1988, pp. 1483-1498. (P. Frankl and Elaine Weyuker)

An Extended Domain-Based Model of Software Reliability, *IEEE Trans. on Software Engineering*, Vol. 14, No. 10, Oct 1988, pp. 1512-1524. [reprinted in *Software Reliability Models: Theoretical Developments, Evaluations, and Applications*, ed. Y.K. Malaiya and P.K. Srimani, IEEE Computer Society Press, 1990, pp. 98-110. (S. Weiss and Elaine Weyuker)

In Defense of Coverage Criteria, *Proceedings of the 11th International Conf. on Software Engineering*, Pittsburgh, Pa, May 1989. (Elaine Weyuker)

What Will It Cost to Test My Software?, *Proceedings of the Seventh Annual Pacific Northwest Software Quality Conference*, Portland, Oregon, September 1989, pp. 349-360. (Elaine Weyuker)

Some Observations on Partition Testing, *Proc. Third Symposium on Testing, Analysis, and Verification*, Key West, Florida, Dec 1989, pp. 38-47. (B. Jeng and Elaine Weyuker)

The Cost of Data Flow Testing: An Empirical Study, *IEEE Trans. on Software Engineering*, Vol. 16, No. 2, Feb 1990, pp. 121-128. (Elaine Weyuker)

Experience With Data Flow Testing, *Proc. Seventh International Conference on Testing Computer Software*, San Francisco, Ca., June 1990, pp. 219-224. (B. Jeng and Elaine Weyuker)