NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE.
UNIVERSITY OF COLORADO

(NASA-CR-163421) SOFTWARE DEVELOPMENT TOOLS: A BIBLIOGRAPHY, APPENDIX C.
(Colorado Univ. at Boulder.) 13 p
HC A02/MF A01
CSCL 093 Unclassified
G3/61 28336

DEPARTMENT OF COMPUTER SCIENCE

Technical Report
SOFTWARE DEVELOPMENT TOOLS:
A BIBLIOGRAPHY

William E. Riddle
Department of Computer Science
University of Colorado at Boulder
Boulder, Colorado 80309

CU-CS-183-80 June, 1980

This work was supported in part, by grant NSG 1638 from NASA Langley Research Center.
SOFTWARE DEVELOPMENT TOOLS:
A BIBLIOGRAPHY

William E. Riddle
Department of Computer Science
University of Colorado at Boulder
Boulder, Colorado 80309

Abstract: We give a bibliography on tools which 1) help software developers perform some development task (such as text manipulation, testing, etc.), and 2) which would not necessarily be found as part of a computing facility. The bibliography comes from a relatively random sampling of the literature and is not complete. But it is indicative of the nature and range of tools currently being prepared or currently available.

This work was supported in part, by grant NSG 1638 from NASA Langley Research Center.
The following citations indicate articles which discuss tools of use in the development of software systems. In compiling this list, we did not include references on:

1. notational tools, that is, languages for the abstract description of software systems,
2. cognitive tools, that is, methods or principles intended to help development practitioners order their tasks and decisions,
3. development environments, that is, collections of tools.

Thus we have focused on augmentive tools, that is, tools which help developers in the performance of some development task such as text manipulation, testing, etc.

We have not included references on "traditional" tools such as compilers, linking loaders, sort routines, file systems, etc. Thus we have focused on tools which one would not pro forma expect to be available under a computing facility. Further, we have included references to techniques which have been developed but not as yet implemented as complete augmentive tools.

This bibliography is not complete. It was augmented somewhat by using the bibliographies prepared as part of term projects by students in the author's graduate-level software engineering course.

It is, however, felt that its randomness is perhaps a strength in that a quick perusal will indicate the nature and breadth of tools currently available.


R.S. Beyer, B. Elspas and K.N. Levitt. SELECT: A formal system for testing and debugging programs by symbolic execution. SIGPLAN Notices, 16, 6 (June 1979), 221-227.


L.W. Cooprider. The representation of families of software systems. 

J.A. Darringer and J.C. King. Applications of symbolic execution to 

J.A. Darringer. The use of symbolic execution in program testing. 
In Software Testing, Infotech International Ltd., Maidenhead, England, 
(1979), pp. 65-84.

G. deBalbine. MTR — A tool for displaying the global structure of 
California, (June 1978), pp. 571-580.

E. Denert. The project library: A tool for software development. 

P. Deutsch. A flexible measurement tool for software systems. Proc. 
IFIP Congress 71, Ljubljana, Yugoslavia, (August 1971), pp. TA-3-7-TA-3-12.

F.M. Djorup and E.T. Irons. A CRT editing system. Comm. ACM, 15, 
11 (January 1972).

International Conf. on Software Engineering, San Francisco, California, 
(October 1976), 182-186.

J.J. Donovan. Tools and philosophy for software education. Comm. ACM, 19, 
8 (August 1976), 430-436.

I.B. Elliott. SPTRAN: A Fortran compatible structured programming 

B. Elspas, K.N. Levitt, R.J. Waldinger and A. Waksman. An assessment of 
techniques for proving program correctness. Computing Surveys, 4, 
2 (June 1972), 97-147.

R.E. Fairley. An experimental program-testing facility. IEEE Trans. on 
Software Engineering, SE-1, 4 (December 1975), 350-357.

R.E. Fairley. Tutorial: Static analysis and dynamic testing of computer 

R.E. Fairley. ALADDIN: Assembly language assertion driven debugging 

R.E. Fairley. Software validation and pre-implementation issues. In 
Riddle and Fairley (eds.), Software Development Tools, Springer Verlag, 
Heidelberg, Germany, (1980).

S.I. Feldman. Make: A program for maintaining computer programs. Bell 

D. Ferrari and M. Liu. A general-purpose software measurement tool. 

A. Fitzsimmons and T. Love. A review and evaluation of software science. 

R.J. Fleischer and R.W. Spitler. SIMON: A project management system for 


D.F. Hare. A structure editor for the Gypsy verification environment. ICSCA-CMP-16, Certifiable Minicomputer Project, Univ. of Texas at Austin, Texas, (July 1979).


P. Henderson. Finite state modelling in program development. SIGPLAN Notices, 10, 6 (June 1975), 221-227.


N. Ng. A graphical editor for programming using structured charts. RJ2344, IBM Research Lab., San Jose, California, (September 1978).


J.L. Peterson. On the formatting of Pascal programs. SIGPLAN Notices, 12, 12 (December 1977), 83-86.


D.J. Reifer. Automated aids for reliable software. SIGPLAN Notices, 10, 6 (June 1975), 131-142.


E.C. Van Horn. Software evolution using the SEER data base. Digital

U. Voges, L. Gmeiner and A.A. von Mayrhauser. SADAT: An automated testing

Third International Conf. on Software Engineering, Atlanta, Georgia,
(May 1978).

G. Wong. Design methodology for computer system modelling tools. In
Zeigler (ed.), *Methodology in Systems Modelling and Simulation*, North-


E. Yarwood. Toward program illustration. CSRG-84, Computer Sys. Research
Group, Univ. of Toronto, Toronto, Canada, (October 1977).


RXVP User’s Guide. RM-1942; General ResearchCorp., Santa Barbara,
California, (1975).

Proceedings 1975 International Conference on Reliable Software, Los Angeles,
California, (April 1975). (Also: SIGPLAN Notices, 10, 6 (June 1975).)

Proceedings First International Conference on Software Engineering,

Proceedings Second International Conference on Software Engineering,
San Francisco, California, (October 1976).

Proceedings Third International Conference on Software Engineering, Atlanta,
Georgia, (May 1978).

Proceedings Fourth International Conference on Software Engineering,
Munich, Germany, (September 1979).
