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.
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.


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. G. Hebalkar. Contour-forming, a mechanism for procedure automation in
the office of the future. RI 2492, IBM Research Lab., San Jose, California,
(March 1979).

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

P. Henderson, R. Gimson, G.D. Pratten and R.A. Snowdon. The maintenance
of software with multiple versions. In Structured System Design, Infotech

P. Henderson. Relational data models for describing system structure.
Computing Lab., Univ. of Newcastle upon Tyne, England.


C.R. Hollander. Decompilation of object programs. TR-54, Stanford

W.E. Howden and L.G. Stucki. Final report: Methodology for effective
test case selection, Phase II. MDCG5800, McDonnell Douglas Corp.,
(April 1975).

W.E. Howden. Experiments with a symbolic execution system. Proc. AFIPS

W.E. Howden. DISSECT: A symbolic evaluation and program testing system.

W.E. Howden. An evaluation of the effectiveness of symbolic testing.
Software: Practice and Experience, 8, 4 (July-August 1978), 381-397.

W.E. Howden. Functional program testing. IEEE Trans. on Software
Engineering, SE-6, (1980).

J.C. Huang. Detection of data flow anomaly through program instrumentation.

C.E. Hughes and C.P. Pfeeger. ASSIST-V: An environment simulator for
IBM 360 Systems software development. IEEE Trans. on Software Engineering,
SE-4, 6 (November 1978).

J. Humbrecht. DEMDAB: A design and maintenance data base system. M.S.
Thesis, Dept. of Computer Sci., Univ. of Colorado at Boulder, Boulder,

J. Hurraas, H. Legard and R.L. Marcus. Pretty: A Pascal pretty printing

D. Ingalls. The smalltalk-76 programming system. Proc. Fifth Annual
ACM Symp. on Principles of Programming Languages, Tucson, Arizona,
(January 1978), pp. 9-16.

F.S. Ingrassia. Combating the 90% complete syndrome. Datamation, 24,
1 (January 1978), 171-176.


J.C. King. A new approach to program testing. SIGPLAN Notices, 10, 6 (June 1975), 228-233.


R.L. London. A view of program verification. SIGPLAN Notices, 10, 6 (June 1975), 534-545.


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.


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


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