

XEROX

**Interlisp-D Reference Manual
Volume I: Language**

3101272
October, 1985

1. The first step in the process is to
 2. determine the requirements of the
 3. system. This is done by
 4. interviewing the user and
 5. analyzing the problem.
 6. The next step is to design the
 7. system. This involves
 8. creating a flowchart and
 9. writing the program code.
 10. The final step is to test the
 11. system. This is done by
 12. running the program and
 13. checking the results.

14. The first step in the process is to
 15. determine the requirements of the
 16. system. This is done by
 17. interviewing the user and
 18. analyzing the problem.
 19. The next step is to design the
 20. system. This involves
 21. creating a flowchart and
 22. writing the program code.
 23. The final step is to test the
 24. system. This is done by
 25. running the program and
 26. checking the results.

27. The first step in the process is to
 28. determine the requirements of the
 29. system. This is done by
 30. interviewing the user and
 31. analyzing the problem.
 32. The next step is to design the
 33. system. This involves
 34. creating a flowchart and
 35. writing the program code.
 36. The final step is to test the
 37. system. This is done by
 38. running the program and
 39. checking the results.

Copyright (c) 1985 Xerox Corporation

All rights reserved.

Portions from "Interlisp Reference Manual" Copyright (c) 1983
 Xerox Corporation, and "Interlisp Reference Manual" Copyright
 (c) 1974, 1975, 1978 Bolt, Beranek & Newman and Xerox
 Corporation.

This publication may not be reproduced or transmitted in any
 form by any means, electronic, microfilm, xerography, or
 otherwise, or incorporated into any information retrieval system,
 without the written permission of Xerox Corporation.

BACKGROUND AND ACKNOWLEDGEMENTS

1 A Brief History of Interlisp

Interlisp began with an implementation of the Lisp programming language for the PDP-1 at Bolt, Beranek and Newman in 1966. It was followed in 1967 by 940 Lisp for the SDS-940 computer, which was the first Lisp system to use software paging techniques and a large virtual memory in conjunction with a list-processing system [Bobrow & Murphy, 1967]. DWIM, the Do-What-I-Mean error correction facility, was introduced into this system in 1968 by Warren Teitelman [Teitelman, 1969].

In 1970 BBN-Lisp, an upward compatible Lisp system for the PDP-10, was implemented under the Tenex operating system [Teitelman, et al., 1972]. With the hardware paging and 256K of virtual memory provided by Tenex, it was practical to provide more extensive and sophisticated user support facilities, and a library of such facilities began to evolve. In 1972, the name of the system was changed to Interlisp, and its development became a joint effort of the Xerox Palo Alto Research Center and Bolt, Beranek and Newman. The next few years saw a period of rapid growth and development of the language, the system and the user support facilities, including the record package, the file package, and Masterscope.

In 1974, an implementation of Interlisp was begun for the Xerox Alto, an experimental microprogrammed personal computer [Thacker et al., 1979]. AltoLisp [Deutsch, 1973] introduced the idea of providing a specialized, microcoded instruction set that modelled the basic operations of Lisp more closely than a general-purpose instruction set could -- and as such was the first true "Lisp machine". AltoLisp also served as a departure point for Interlisp-D, the implementation of Interlisp for the Xerox 1100 Series of personal computers, which was begun in 1979 [Sheil & Masinter, 1983].

In 1976, partially as a result of the AltoLisp effort, a specification for the Interlisp "virtual machine" was published [Moore, 1976]. This attempted to specify a small set of "primitive" operations

which would support all of the higher level user facilities, which were nearly all written in Lisp. Although incomplete and written at a level which preserved too many of the details of the Tenex operating system, this document proved to be a watershed in the development of Interlisp, since it gave a clear definition of a (relatively) small kernel whose implementation would suffice to port Interlisp to a new environment. This was decisive in enabling many subsequent implementations.

Most recently, the implementation of Interlisp on personal workstations has extended Interlisp in major ways. Most striking has been the incorporation of interactive graphics and local area network facilities. Not only have these extensions expanded the range of applications for which Interlisp is being used, but the personal machine capabilities have had a major impact on the Interlisp programming system itself. Whereas the original Interlisp user interface assumed a very limited (teletype) channel to the user, the use of interactive graphics and the "mouse" pointing device has radically expanded the bandwidth of communication between the user and the machine. This has enabled completely new styles of interaction with the user (e.g., the use of multiple windows to provide several different interaction channels with the user) and these have provided both new programming tools and new ways of viewing and using the existing ones. In addition, the increased use of local area networks (such as the Ethernet) has expanded the horizon of the Interlisp user beyond the local machine to a whole community of machines, processes and services. Large portions of this manual are devoted to documenting the enhanced environment that has resulted from these developments.

2 Interlisp Implementations

Development of Interlisp for the PDP-10 was, until approximately 1978, funded by the Advanced Research Projects Administration of the Department of Defence (DARPA). Subsequent developments, which have emphasized the personal workstation facilities, have been sponsored by the Xerox Corporation, with contributions from members of the Interlisp user community.

Although there are a variety of implementations of Interlisp in use, this manual is a reference manual for the Interlisp-D implementation. Notes may occasionally be included on other implementations, but there is no guarantee that this information is complete for implementations other than Interlisp-D. For some implementations, there is a "Users Guide" which documents features which are completely unique to that

machine; for example, how to turn on the system, logging on, and unique facilities which link Interlisp to the host environment or operating system.

3 Acknowledgements

The Interlisp system is the work of many people -- after nearly twenty years, too many even to list, much less detail their contributions. Nevertheless, some individuals cannot go unacknowledged:

Warren Teitelman, more than anyone else, made Interlisp "happen". Warren designed and implemented large parts of several generations of Interlisp, including the initial versions of most of the user facilities, coordinated the system development and assembled and edited the first four editions of the Interlisp reference manual.

Larry Masinter is a principal architect of the current Interlisp system, has contributed extensively to several implementations, and has designed and developed major extensions to both the Interlisp language and the programming environment.

Dan Bobrow was a principal designer of Interlisp's predecessors, has contributed to the implementation of several generations of Interlisp, and (in collaboration with others) made major advances in the underlying architecture, including the spaghetti stack, the transaction garbage collector, and the block compiler.

Ron Kaplan has decisively shaped many of the programming language extensions and user facilities of Interlisp, has played a key role in two implementations and has contributed extensively to the design and content of the Interlisp reference manual.

Peter Deutsch designed the AltoLisp implementation of Interlisp which developed several key design insights on which the current generation of personal machine implementations depends.

No matter where one ends this list, one is tempted to continue. Many others who contributed to particular implementations or revisions are acknowledged in the documentation for those systems. Following that tradition, this manual, which primarily documents the Interlisp-D implementation, acknowledges, in addition to those listed above, the work of:

Bill van Melle, who designed and implemented most of the local area network facilities, the process mechanism, and much of the run time support system.

Richard Burton, who designed and implemented a great deal of the interactive display facilities.

and the contributions of Alan Bell, Don Charnley, Mitch Lichtenberg, Steve Purcell, Eric Schoen, Beau Sheil, John Sybalsky, and the many others who have helped and contributed to the development of Interlisp-D.

Like Interlisp itself, the Interlisp Reference Manual is the work of many people, some of whom are acknowledged above. This edition was substantially rewritten, designed, edited and produced by Michael Sannella of Xerox Artificial Intelligence Systems. It is a major revision of the previous edition --- it has been completely reorganized, updated in most sections, and extended with a large amount of new material.

Interlisp is not designed by a formal committee. It grows and changes in response to the needs of those who use it. Contributions and discussion from the user community remain, as they always have been, warmly welcome.

4 References

- | | |
|---------------------------|--|
| [Bobrow & Murphy, 1967] | Bobrow, D.G., and Murphy, D.L., "The Structure of a LISP System Using Two Level Storage" --- <i>Communications of the ACM</i> , Vol. 10, 3, (March, 1967). |
| [Deutsch, 1973] | Deutsch, L.P., "A Lisp machine with very compact programs" --- <i>Proceedings of the Third International Joint Conference on Artificial Intelligence</i> , Stanford, (1973). |
| [Moore, 1976] | Moore, J.S., "The Interlisp Virtual Machine Specification" --- Xerox PARC, CSL-76-5, (1976). |
| [Sheil & Masinter, 1983] | Sheil, B., and Masinter, L.M. (eds.), "Papers on Interlisp-D" --- Xerox PARC, CIS-5 (Revised), (1983). |
| [Teitelman, 1969] | Teitelman, W., "Toward a Programming Laboratory" --- <i>Proceedings of the International Joint Conference on Artificial Intelligence</i> , Washington, (1969). |
| [Teitelman, et al., 1972] | Teitelman, W., Bobrow, D.G., Hartley, A.K. Murphy, D.L., <i>BBN-LISP TENEX Reference Manual</i> --- Bolt Beranek and Newman, (July 1971, first revision February 1972, second revision August 1972). |
| [Thacker, et al., 1979] | Thacker, C., Lampson, B., and Sproull, R., "Alto: A personal computer" --- Xerox PARC, CSL-79-11, (August, 1979). |