SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

LEVEL II

1. REPORT NUMBER: FINAL	2. GOVT ACCESSION NO. (10)
4. TITLE (and Subtitle): (6) Research on Interlisp and Network-Based Systems.	(9) FINAL rept. 1 Jul 77 - 31 Mar 78
7. AUTHOR(s): (10) W./Teitelman	8. CONTRACT OR GRANT NUMBER(s): (15) N00014-77-C-0277, ARPA Order-2906
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Xerox Corporation (New) 3333 Coyote Hill Road Palo Alto, California 94304	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR 049-372/12/29/76 Order No. 2906
11. CONTROLLING OFFICE NAME AND ADDRESS: -	12. REPORT DATE: 5/24/78
	13. NUMBER OF PAGES: 3 (11) 24 May 78
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): -	15. SECURITY CLASS. (of this report): Unclassified (12) 14p.
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N.A.

16. DISTRIBUTION STATEMENT (of this Report)

Unlimited Distribution

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Unlimited Distribution

18. SUPPLEMENTARY NOTES

None

DDC RECEIVED JUL 24 1978 B

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

-

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The contract covered by this final report ~~includes~~ covers a variety of activities and services centering around the continued growth and well-being of Interlisp, a large interactive system widely used in the ARPA Community for developing advanced and sophisticated computer-based systems.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 2906

ARPA Order No. 2906 Contract Number: N00014-77-C-0277
Program Code No. 3M600 Principal Investigator: W. Teitelman
Xerox Corporation (415) 494-4447
Effective Date: 1 January, 1977 Title: INTERLISP
Expiration Date: 31 March, 1978
Amount of Contract: $60,000 Date of this report: May 24, 1978

1 July, 1977 - 31 March, 1978

Research Program Plan

a) Maintenance -- Maintenance of machine-independent portion of Interlisp, finding and fixing bugs, and modifying or improving exisitng features.

b) Documentation -- producing and maintaining documentation for Interlisp, including an on-line Interlisp reference manual.

c) Distribution -- periodically bringing up and checking out new releases of Interlisp-10, and distributing these releases to selected ARPA sites.

d) Coordination -- insuring that the various user communities are kept informed of developments in Interlisp, and that they have the opportunity to participate in and influence such developments, insuring that new implementations of Interlisp conform to the Interlisp virtual machine. Providing consultation and assistance to implementors. Making changes to Interlisp where necessary to assist other implementors.

Major Accomplishments

Builders of large systems in Interlisp commented that they often had numerous files that were needed only when they were going to do further developement, but not by ordinary users, e.g. files containing various record declarations, macro and other properties, etc. The designers of these systems wanted to maintain only one set of files, and somehow be able to control at load-time what happened, i.e. to have a conditional load operation which in one context performed a certain set of operations at load time, and in another context perhaps an entirely different set of operations. Therefore, we extended the semantics of the load and compiler directives to permit specifying such conditional operations.

Similarly, to facilitate efficient loading of systems built in Interlisp, a SYSLOAD option was added to the Interlisp load capability which takes notice of the fact that the file in question is being loaded for production, not for developement, and disables various Interlisp features such as saving a file map, noticing the file for the file package, making the load be undoable etc., with a

concomitant savings storage in the resulting system.

The BREAKDOWN package for measurement and performance evaluation was extended to permit the user to specify measuring several quantities simultaneously, e.g. computation time and page faults. In addition, for more accurate measurements, we added an option for specifying that the actual measurement operation be performed in-line rather than via a call to a separate function.

In collaboration with BBN, we cleaned up and respecified the error conditions with respect to opening files. For example, previously a FILE NOT FOUND error would be generated when in fact the file existed but would not open due to protection violation or some other cause. Similarly, a FILE WON'T OPEN error occurring could mean there was no space left in the system, or because another fork had the same file open. We have added three new error conditions to the two mentioned above: FILE SYSTEM RESOURCES EXCEEDED, PROTECTION VIOLATION, and BAD FILE NAME. It is now possible for a user to determine when a file does not open exactly why.

In the interest of human engineering, as well as making Interlisp more implementation independent, all dependence on type numbers was removed from the system and replaced by type names. For example, the garbage collector now prints "collecting string pointers" whereas before it printed GC: 30. Similarly, any functions which deal with data types, such as RECLAIM, MINFS, DEFPRINT, etc., now all deal with type names, and the function STORAGE now prints out descriptive information rather than type numbers.

In a similar vein, the control-T feature provided by the operating system (Tenex or Tops-20) which is used to interrogate the state of a program, e.g. is it running, in an iowait, etc., (or, in the absence of a response, to determine that the time-sharing has, in fact, crashed!), was replaced by a similar feature that is more Lisp oriented: instead of reporting the program location currently being referenced, e.g. RUNNING AT 20637, the new control-T feature gives a mini-backtrace of the computation, e.g. RUNNING IN EVAL IN LOAD IN LISPX. Thus, the user can determine where the program is spending its time, and occasionaly even spot a malfunction by the fact that the system is executing functions that it should not be calling in the given context.

The BREAKRESETFORMS facility was designed to allow users to debug those parts of their programs that would otherwise interact with the debugging machinery, e.g. input routines, by permitting the user to specify operations to be performed before a break occurred, and then to be reversed after the break was completed, thereby both isolating and capturing the object program's environment. We have extended this facility to permit saving and restoring values of system parameters. We also made the BREAKRESETFORMS facility more robust in handling those situations where an infinite loop would have occurred as a result of a break being on some function that was itself invoked via BREAKRESETFORMS.

We implemented a significantly improved version of two frequently used Interlisp primitives, SUBLIS and SUBPAIR, which are both much faster than the old implementations, as well as using only one third of the stack space required previously, thus allowing the size of the object being substituted into to be three times as large.

The spelling corrector was improved with respect to handling errors involving upper/lower case names. In particular, when the user types in an atom that is correct except for the fact that it is in lower case where the system expects upper case, the correction is performed immediately, and without any interaction with the user.

Due to DEC's decision to make a number of backwards incompatible changes in release 3 of TOPS-20 operating system, in particular, that user numbers and directory numbers would no longer be the same, it was necessary to redesign and reimplement the user profile-initialization facility of Interlisp in order that it continue to run at both Tenex and Tops-20 sites.

During this period, we installed or helped to install versions of Interlisp at six sites: I4-TENEX, Caltech TOPS-20, Yale TOPS-20, Carnegie Mellon Tenex, University of Utah TOPS-20, Columbia University TOPS-20, and Stanford (SU-AI). This involved transferring the relevant Interlisp source and system files to the site, either over the ARPA net, or by making a tape and sending it through the mail, instructing local personnel as to how to obtain and run new versions of Interlisp, and in some cases, performing certain patches to the TENEX operating system in order to make it compatible with TOPS-20 operating system with repesect to the instruction used by Interlisp to determine dynamically which system it is running under. The continued growth of the Interlisp community is partly due to the quality of the support and maintenance that the Interlisp system receives under the ARPA contracts with PARC and BBN.

A number of bugs were reported to PARC that, when analysed, were found to fall under the purview of BBN. However, PARC personnel invested significant amounts of time in isolating and locating these bugs, and in some cases implementing fixes which were later accepted by BBN and incorporated into their portion of the Interlisp system. These included bugs in the garbage collector, the swapper, and in the handling of shadow forks.

In addition to the above, we received and responded to a variety of reports of bugs, non-features, suggestions for additions or changes, or simply requests for information, from sixteen ARPA sites: SRI-KL, MIT-AI, USC-ECL, USC-ISIE, BBN-TENEXD, SU-AI, USC-ISI, USC-ISIB, BBN-TENEXA, SUMEX-AIM, I4-TENEX, USC-ISIC, CMU-10A, MIT-MC, CMU-10B, UTEXAS, RUTGERS-10.

During this period, there were eight new releases of INTERLISP.