PREVIOUS SPONSORS OF LISP POINTERS

IBM Thomas J. Watson Research
Institut National De Recherche En Informatique Et En Automatique (I.N.R.I.A.)
Xerox PARC
Microelectronics and Computer Technology Corporation (MCC)
Texas Instruments
Digital Equipment Corporation
Mitre Corporation
ACM SIGPLAN
Sun Microsystems
Lucid, Inc.

Dear Colleague,

By now you should be really confused by the dates on this volume. We've decided that we're going to follow the precedent of most of the ACM newsletters and use a Calendar Year volume dating rather than a Fiscal Year. This means that Volume IV, Number 1 will now start in January, 1991. Showing July through December, 1990 is just to get our calendars adjusted. If you think YOU'RE confused...

Although we are no longer showing our original sponsors, I'd like to thank all of them again for their help in bringing us this far. We will be looking into newsletter sponsorship at some time in the future, so don't think you're completely safe from us forever.

So, this is now our second year under ACM SIGPLAN and the first year that our subscribers will have to pay for their subscriptions. We do intend to make it a good year for you. You can help by your contributions to our various departments. We always need material and you will find help and support in the department chairs. Something we can always use are reviews - whether of books or videotapes. Of course, if you are interested in taking on a department, just call and ask.

I would also like to ask you to think back to your years in the field and see if there's some personal story that you might like to send us for its historical value. I've just finished shooting a documentary on John Cocke, a computer architect at IBM, and my consciousness is most definitely raised. Someday, someone is going to want to piece together various stories of how things happened. Newsletters, like this, can provide a place to find some of that kind of information. So, go back over your dusty shelves and see if your memory is stirred by people you worked with and projects that happened and try to share some of those memories with us.

This is a very large issue. Much of the thanks go, again, to JonL White for putting together a very fine group of theme papers on Lisp-based UI Tools. You will also notice that we have a new associate editor, Judy Chien, who will be taking over from Bonnie Zingerelli. Thanks to both for their help.

Sincerely,

Mary S. Van Deusen
Editor

From: Alan@AI.AI.MIT.EDU (Alan Bawden)
Subject: self reproducing code
Date: 7 Oct 88 20:00:00 GMT

> Date: Fri, 7 Oct 88 13:30:30 MDT
> From: carr%car@cs.utah.edu (Harold Carr)
> A couple of years ago I found an example of self reproducing code....
> in Common Lisp, ...

My favorite example:

```
(let ((let '*(let ((let ',let))
                ,let)))
  *(let ((let ',let))
     ,let))
```

I believe that Mike McMahon is the author of this variant. It is unfortunate that this isn't standard Scheme because of the use of LET as an identifier...

NOTICES

EUROPAL '91
The Second European Conference on the Practical Applications of LISP, Europal '91, will be held at Churchill College, Cambridge, England. March, 1990. For information, contact Europal '91, P.O. Box 110, Dorking, Surrey, RH5 4FB, United Kingdom, Phone: +44 (0) 306 77331, Fax: +44 (0) 306 77696.

SPECIAL ISSUE: Lisp-Based UI Tools

Dear Lisp Friends:

In this issue of LISP POINTERS we have three excellent papers describing the building of user-interface and windowing tools with Lisp technology. And what better demonstration of the advantages of dynamically-typed, user-extensible programming languages! Back in the mid 1970's, SmallTalk was making its mark by showing off flashy windowing and graphics-like demos -- at a time when the Model 35 TeleType was still not an unforgotten beast! It is no accident that the dynamic capabilities of SmallTalk and Lisp are very similar, from automated memory management (garbage collection) to runtime type dispatching; and in these three papers, we will see a little bit more about how liberating these capabilites are when carrying out almost any UI development.

The integration into the Lisp world of object-oriented ideas began very shortly after the mid-1970's, and was spurred principally by the desire to have a comparable User Interface on the experimental Lisp machines being developed then with bit-mapped raster displays, as opposed to the usual 24x80 column terminals. Of course, there were research efforts in programming languages during this timeframe, some of which had object-oriented parts, but the SmallTalk windowing play-pen was a "bug" that bit the Lisp world in a big way back then. The group working on the MIT Lisp Machine soon found itself emulating many of SmallTalk's features, such as classes, methods, inheritance and message-passing in order to build useful windowing capabilities; and similar work was going on at Xerox's Palo Alto Research Center in the parts of the lab that were porting Interlisp off the PDP10 computers and onto the same kinds of bit-mapped workstation machines that SmallTalk had been so successful on.

During 1979 through 1981, the MIT world developed a multiple-inheritance extension to Lisp called FLAVORS, and the window system for the Lisp machine was completely rewritten in it, using its unique paradigm of "mixin" classes. On the west coast, the PARC group developed an AI language called LOOPS (Lisp Object-Oriented Programming System), in which the development environment facilitated the kinds of tweaking, modification, and "re-design on the fly" that have so long been characteristic of the AI and Lisp worlds, but which are now beginning to be recognized as crucial to the shaping and use of interactive User Interfaces.

Quite naturally, then, we find CLOS, the spiritual heir to FLAVORS and LOOPS, being showcased in two of these three user-interface papers. And the third? WINTERP? Well, you'll be surprised to see how a little bit of object-orientedness can go a long way ***in a dynamically-typed and interpreted*** world; and you'll be even more surprised to see how effective even a "little" lisp can be in solving huge UI problems. So, try a little lisp—you'll find it's good for you.

In another vein, let me encourage you again to send us articles of any interesting nature about the Lisp world. For several years, we have been publishing research-like papers, and language design papers. But it may be high time to see some tutorial-like monographs, or some papers summarizing areas of discussion and ferment, relevant to the Lisp world. There were dozens of attendees at the CLOS workshop held in conjunction with the ECOOP/OOPSLA '90 Conference last October in Ottawa; and there are hundreds (perhaps thousands) of observers monitoring the various electronic mailing lists connected in one way or another with Lisp. Perhaps it's time for some of you readers who have thoughts and serious question about these developments to put a pen to them, and to send the result to us an issue-summary paper, or as tutorial paper.

Cheers, and we look forward to hearing from you!

Jon L White
Technical Articles Editor
Lucid, Inc.
707 Laurel Street
Menlo Park CA 94043

Internet: jonl@lucid.com

How can SIGPLAN help you?

Do you want to involve yourself in standards work, but need financial support? Are you a student who has a paper to present, but needs money to attend the conference? The SIGPLAN Professional Activities Committee (PAC) may be able to help you. PAC exists to provide a source of *limited financial support* for ACM and SIGPLAN members to advance their professional goals. A limited amount is budgeted for PAC activities in any one year. Applications are treated on a first-come, first-serve basis. PAC expects to fund applications for (in order of priority):

- annual fees $$$$ or limited travel expenses for attending a standards or other voluntary committee meeting

- registration fees, accommodations and/or travel expenses for student attendance at a workshop or conference where the applicant is to present a paper; (PAC encourages institutions to help defray the cost of funding students.)

- any other activity deemed appropriate by the committee

It is ***essential*** that applicants pay strict attention to the application submission requirements listed below, since failure to do so delays consideration of the application. Applications should be made ahead of time, in particular to permit taking advantage of inexpensive fares. It is **recommended** applications be sent at least **three months** before the activity is to begin (or **two months** for supported attendance at SIGPLAN-sponsored conferences).

The committee listed below will review applications and make recommendations to the Chair of SIGPLAN who finalizes the awards. Applications must include:

1) a resume

2) a description of the activity and its relevance to SIGPLAN

3) a budget including all available and prospective support, with an endorsement from the applicant's employer or institution (e.g., department chair)

4) a statement concerning official representation (NOTE: Each applicant is required to include a statement with the application stating **either** that he/she has been designated an official representative by the ACM Council **or** that the applicant does not act in any official capacity as a representative of ACM or SIGPLAN. Clearly, this is not relevant in the case of student attendance at conferences.)

Applications should be submitted **in parallel,** and preferably by e-mail, to:

Barbara Ryder
Dept. of Computer Science
Hill Center, Busch Campus
Rutgers University
New Brunswick, NJ 08903, USA
ryder@cs.rutgers.edu

Julian Padget
Laboratory for Computer Science
University of Bath
Bath, Avon, BA2 7AY
United Kingdom
jap@maths.bath.ac.uk

Mary Lou Soffa
307 MIB
Computer Science Dept.
University of Pittsburgh
Pittsburgh, PA 15260, USA
soffa.@cs.pitt.edu

Call for Papers
ACM SIGPLAN'91 Conference on
Programming Language Design and Implementation
Toronto, Ontario, Canada, June 26-28, 1991

SIGPLAN'91 continues the series of broad-based language and compiler design conferences. The conference will provide a forum for researchers and developers to gain awareness of current practical and experimental work across the breadth of the field. Emphasis will be placed on experimental results and experience with the languages and techniques described.

The conference seeks original papers relevant to practical issues concerning the design, development, implementation, and use of programming languages (in contrast to the annual SIGACT/SIGPLAN POPL Conference which is oriented more toward foundations). The conference does not favor any particular programming paradigm or support architecture. As usual for SIGPLAN-sponsored conferences, papers awaiting acceptance to any other conference are not eligible for SIGPLAN'91; if a closely related paper has been submitted to a journal, the program chair must be notified.

Among the topics encompassed by the conference are:

- compiler construction
- interpretation
- implementation by preprocessing
- design and use of languages
- special-purpose languages
- programming environments
- translation by program transformation
- benchmarks and assessment
- translator validation
- implementation for non-traditional languages and/or architectures
- design of internal representations
- incremental and interactive methods
- optimization techniques for scalar and parallel architectures

As usual, papers will be selected on the basis of novelty, significance, and overall quality of the contributions. Submissions should be in the form of an extended abstract summarizing the major results to be presented, their importance, and their relationship to other work in the field. The submissions will be read and evaluated by the program committee and should be sufficiently complete to be the basis for selection. The extended abstract should be **no more than** ten pages, typed double spaced or typeset 10 point type on 16 point spacing.

Authors should submit twelve copies of an extended abstract (double sided, if possible) to be received by the program chair no later than **November 19, 1990**. Persons submitting papers from countries in which access to copying machines is difficult or impossible may submit a single copy. Authors will be notified of acceptance or rejection by **January 28, 1991**. The accepted papers must be formatted according to ACM conventions and be received by the program chair no later than **March 18, 1991**. Authors of accepted papers will be expected to sign an ACM copyright release form. Proceedings will be distributed at the conference and as a special issue of SIGPLAN Notices. All papers published in Proceedings are eligible for publication in refereed ACM publications at the discretion of the editor.

Program Committee:

Andrew Black, *DEC Cambridge Research Laboratory*
Christopher W. Fraser, *AT&T Bell Laboratories*
Peter Lee, *Carnegie Mellon University*
Steven S. Muchnick, *SUN Microsystems*
Barbara Ryder, *Rutgers University*
Leon Sterling, *Case Western Reserve University*
Kent Dybvig, *Indiana University*
James Larus, *University of Wisconsin - Madison*
Jean-Jacques Lévy, *INRIA-Rocquencourt*
William Pugh, *University of Maryland*
Vivek Sarkar, *IBM T J Watson Research Center*
William E. Weihl, *MIT*

The conference will be preceded by two days of tutorials on June 24-25. There will be two tracks as in previous years: the first will present an overview of the fundamentals of compiler construction for languages such as FORTRAN, Pascal, and C; the second will consider advanced topics.

Conference Chair:
Brent Hailpern
IBM
Old Orchard Road
Armonk, NY 10504
(914)765-6481
bth@ibm.com

Program Chair:
Barbara Ryder
Department of Computer Science
Hill Center, Busch Campus
Rutgers University
New Brunswick, NJ 08903
(201)932-3699
ryder@cs.rutgers.edu

Local Arrangements Chair:
Shahram Javey
IBM Canada Ltd., Dept.3T-520-1150-TOR
1150 Eglinton Ave. East
North York, Ontario M3C 1H7
Canada
(416)448-4201
javey@ibmlabnn.bitnet