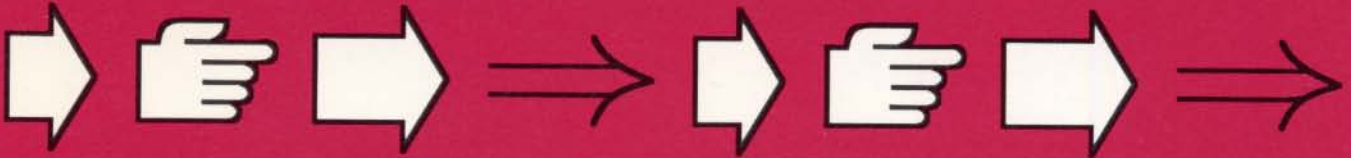


LISP



Pointers



SIGPLAN Special Interest Publication on LISP

Volume 3, Number 1

July-December 1989 January-March 1990

TABLE OF CONTENTS

Letter From Editor, Van Deusen	1
Letter From Assoc. Editor, Zingarelli	2
Letters to the Editor, Camp	3
1990 ACM Reminder	5
Technical Articles, White	6
The Series Macro Package, Waters	7
Concise Reference Manual for the Series Macro Package	12
Standard-Output, Steele	29
SIGPLAN - PAC	34
Programming Environments, Foderaro	
UnicStep-a Visual Stepper for COMMON LISP, Haulsen and Sodan	35
Call for Papers - ECOOP-OOPSLA '90	44
Lisp and Symbolic Computation Journal	45
International News, A Subjective View of Lisp, Quinnc	46
(algorithms), Pavel	48
Readings in Scheme, Yigit	62

EDITORIAL BOARD

EDITOR

Mary S. Van Deusen
IBM Watson Research
P.O. Box 704
Yorktown Heights, NY 10598
(914) 784-7845 (**Note change)
(914) 232-7490
(508) 384-2526
maida@ibm.com

ASSOCIATE EDITOR

Bonnie F. Zingarelli
IBM Watson Research
P.O. Box 704
Yorktown Heights, NY 10598
(914) 784-7966
zinger@ibm.com

TECHNICAL ARTICLES

JonL White
Lucid, Inc.
707 Laurel Street
Menlo Park, CA 94025
(415) 329-8400 X5514
edsel!jonl@labrea.stanford.edu

*** (*STANDARD-OUTPUT*) ***

Guy Steele
Thinking Machines Corporation
245 First Street
Cambridge, MA 02142
617-876-1111
Internet gls @ think.com

PROGRAMMING ENVIRONMENTS

John Foderaro
Franz Inc.
1995 University Avenue #275
Berkeley, Ca. 94704
(415) 548-3600
jkf%franz.uucp@berkeley.edu

*** (*QUERY-IO*) ***

Patrick Dussud
Lucid, Inc.
707 Laurel Street
Menlo Park, CA 94025
(415) 329-8400 X5578
dussud@lucid.com

INT'L NEWS

Christian Queinnec
LIX Laboratoire d'Informatique de
l'Ecole Polytechnique
Route de Saclay
91128 Palaiseau Cedex
FRANCE
E-mail: queinnec@poly.polytechnique.fr
UseNet: mcvax!inria!poly!queinnec

ALGORITHMS

THE SCHEME OF THINGS

Pavel Curtis
Xerox PARC
3333 Coyote Hill Rd.
Palo Alto, CA 94304
(415) 494-4455
Pavel.pa@Xerox.Com

READINGS IN SCHEME

Ozan S. Yigit
York University
Dept. of Computing Services
4700 Keele Street, North York
Ontario, M3J 1P3
CANADA
(416) 736-5257 X3976
oz@nexus.yorku.ca

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,

My deepest apologies for being so late with this issue. Exhaustion is my only excuse. To help with this, I am getting the aid of an Associate Editor. Bonnie Zingerelli will be handling the assembly of the issues. Feel free to send material to either one of us. I'll see your submissions in either case.

We are now an official technical publication of the ACM SIGPLAN (Special Interest Group in Programming Languages). SIGPLAN has agreed to make four issues of Lisp Pointers available to our old subscribers for free. Since these four issues have to be out by June 31st, 1990, you might find our third and forth issues combined into a single large special issue.

An old subscriber is one who was on our mailing list as of the time we put out this issue. If you want to become a subscriber now, you will have to fill out the form in the rear of this issue and subscribe through ACM. You will notice that the rate is still very low. This represents a subsidy by SIGPLAN of a subscribers membership rate.

An old subscriber can update their address by sending an address change to Bonnie Zingerelli. We will stop maintaining the old mailing list after Volume III, Number 4's labels have been printed.

Back issues of Lisp Pointers have been sent to ACM Headquarters and requests for back issues should be made through them.

There are some changes in our departments with this issue. Bob Mathis' "X3J13" department has become *standard-output* and is now being headed by Guy Steele. Will Clinger is giving up his department to Pavel Curtis who will be taking on BOTH "The Scheme of Things" and "(Algorithms)". Our thanks to Pavel for dauntless bravery and to Bob and Will for all the hard work they have devoted to this publication over the last two volumes. We'll be seeing more of Will as the editor of a special issue of Lisp Pointers devoted to the Scheme standard. Unfortunately, Walter did not hear the sound of clapping hands and "Implementation Summaries" has died. This is a great loss as Walter put out a truly wonderful department. If anyone has ideas for a new department, and the time and energy to volunteer, we'd like to hear from you. This publication is only what you make of it.

I should remind you that technical articles should be submitted to JonL. White unless they fit specifically into one of the areas described by a particular department. The strength of this publication is due to the high quality of these department chairs. All articles are reviewed for acceptance. If you have an idea for a submission, send it to one of them. They're anxious to hear from you.

We're keeping the style of our old cover designed by Cyril Alberga with a new update by Randall Koons. The new cover will be Cambridge Cranberry -- in memory of Lisp work past.

Our thanks also to the sponsors who have kept us going for these last two volumes. We'll be keeping your names on our publication throughout this next volume to remind our subscribers of your kindness in supporting us. Without you, there wouldn't be an us.

Sincerely,



Mary S. Van Deusen
Editor

Greetings!

Along with other changes commencing with this publication of LISP Pointers which I'm sure you have already begun noticing, there is now an Associate Editor-me! I'll be collecting articles from the various departments, putting together the physical newsletter, and sending mailing labels to ACM for the issues which our old members will be getting for free. I won't be taking any new subscriber information (see the inside rear cover for that), but I will be taking address changes.

I'm looking forward to this new venture and would appreciate comments from you - our devoted readers - on any subject. Remember-this is your publication and to make it work for you, we need your input! My address is right in the front.

Sincerely,



Bonnie F. Zingarelli
Associate Editor

```
*** ERROR MESSAGES I REMEMBER
*** (collected at June, 1989 X3J13 meeting)
```

```
Gleep - out of bit blocks
```

```
;BKPT BARF - CALL JONL (from PDPIC MacLisp Compiler - Jon L. White)
```

```
;VERITAS AETERNA - DON'T SETQ T (MacLisp - Barry Margolin, Guy Steele)
```

```
;NILIL EX NIHIL - DON'T SETQ NIL
```

```
;PURITAS NELESSE EST - DON'T DO RANDOM BINDINGS
```

```
#<Unprintable object:#<Unprintable object:#<Unprintable object:#<...
(unnamed system)
```

```
Window System Locked -- Using the Cold-Load Stream
(Lisp Machine window system)
```

```
;TOO MUCH MUSIC (Lisp LOGO, 1974) (refers to music buffer overflow)
```

```
Fatal Error in DRIBBLE: Double dribbling is not allowed
(early version of VAX LISP, when trying to nest DRIBBLEing)
```

```
Core dumped. %
```

```
The only metaclass STANDARD-CLASS class that a built-in class can have as a
super is the class T.* Error message from PCL, 1988.
```

“Jumpy” and “Pushy” call/cc

TANAKA Tomoyuki

IBM Research, Tokyo Research Laboratory
5-19, Sanbancho, Chiyoda-ku, Tokyo 102, Japan
e-mail: `tanakat@jpntscvm.bitnet`

16 January 1990

I greatly enjoyed Queinnec and Séniak’s (`call/cc call/cc`) puzzles in the last issue of *Lisp Pointers*. [Queinnec and Séniak 89] I would like to clarify what they mean by “jumpy” and “pushy” `call/cc`.

The terms *jumpy* and *pushy* that Danvy and Malmkjær introduced in [Danvy and Malmkjær 88] refer to how continuations are applied in the presence of tower of processors. When a reified continuation is applied in a *jumpy* way, the continuation of the application form is discarded. When a reified continuation is applied in a *pushy* way, the continuation of the application form is stacked on the meta-continuation. Therefore, if there is a method for *exiting* to the continuation at the top of the meta-continuation, the stacked continuation can be reactivated.

Normally `call/cc` is defined (in Brown [Friedman and Wand 84] notation) as:

```
(abs simple (f)
  ((abs reify (e r k) (k (f k))))))
```

In this definition the reified continuation may be applied in either jumpy or pushy way. Without an exiting mechanism, however, this makes no observable difference, for there would be no way to retrieve the continuation stacked on the meta-continuation.

What Queinnec and Séniak mean by a “jumpy” `call/cc` is related but different from the above usage. A “jumpy” `call/cc` is defined as:

```
(abs simple (f)
  ((abs reify (e r k) (wrong (f k))))))
```

where `wrong` is a “black hole” continuation which signals an error.¹ This is different from an ordinary `call/cc` in that the argument function must explicitly invoke the captured (reified) continuation; otherwise an error is signalled. For example, with an ordinary `call/cc`:

```
(call/cc (lambda (c) (c 1))) ==> 1
(call/cc (lambda (c) 1)) ==> 1
```

With a “jumpy” `call/cc`:

```
(call/cc (lambda (c) (c 1))) ==> 1
(call/cc (lambda (c) 1)) ==> <<an error is signalled>>
```

I believe that this variant of `call/cc` is new to the Lisp community.

References

[Danvy and Malmkjær 88]

Danvy, O., and Malmkjær, K. Intensions and Extensions in a Reflective Tower. In *Proceedings of the 1988 ACM Conference on Lisp and Functional Programming*. 1988, pp. 327–341.

[Friedman and Wand 84]

Friedman, D.P., and Wand, M. Reification: Reflection without Metaphysics. In *Proceedings of the 1984 ACM Conference on Lisp and Functional Programming*. 1984, pp. 348–355.

[Queinnec and Séniak 89]

Queinnec, C., and and Séniak, N. Puzzling with Current Puzzle. *Lisp Pointers*. Vol. 2, No. 3–4 (1989), pp. 4, 14, and 39.

¹`wrong` may alternatively be defined to go into an infinite loop.

*** Reminder ***

The 1990 ACM Conference
on Lisp and Functional Programming
will be held June 27-29, 1990
in Nice, France!

For more information contact:

Gilles Kahn
INRIA Sophia Antipolis
2004 Route DES LUCIOLES
Valbone, CEDEX 06565

August, 1989

Dear Lisp Friends:

Welcome again to our forum for technical interchange. We are especially happy with SIGPLAN support of Lisp Pointers, and hope you will all join us in continuing to make the effort a successful one.

Lisp Pointers provides a rapid-turnaround channel for technical articles that requires more thought and preparation than a typical electronic bulletin-board message, but is less stringent than the time-consuming referee and review process typical of most academic journals. Note that the authors of articles published herein retain the copyright to their material, so that they may republish it later wherever they choose.

We solicit your participation by asking you to write down your interesting thoughts, ideas, and articles related to theory, use, design, monitoring and programming in the Lisp language. Then send them to us! Remember, this call is for "interesting" papers—not necessarily something as deep or complicated as a Ph.D. thesis. Our focus in Lisp Pointers is on the pragmatic side, but we want to encourage discussion of new, and even untested, ideas which will likely have some practical interest.

In this issue, we are featuring a novel proposal for an alternative to iterative style programming—and it isn't just recursion style! Richard Waters, of MIT, has been working on this approach for many years now, and it has reached some degree of maturity.

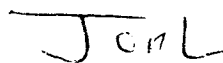
Research into all areas of Lisp development, performance monitoring, and language design would be welcome, whether original or not. Lengthy monographs would not be appropriate for submission here, but selected parts are most welcome, keeping in mind the page limitation; this would include descriptions of partial results which are in the process of being developed for eventual use elsewhere, such as in books, theses, journals and so on. Especially welcome would be non-research articles such as tutorial-like coverages of current topics, and coherent summaries of topics that have been actively discussed on the various electronic mailing lists. Useful Lisp programs, or annotated algorithms, should be sent directly to the Algorithms editor.

The physical layout of submissions should adhere to the policy stated at the back of this newsletter, with an expected length of five to ten pages (for exceptions to the length restriction, contact the Technical Articles Editor directly). Transmittal may be either the camera-ready hardcopy mailed to the address below, or Tex or LaTeX sources electronically mailed through the ArpaNet; sources may also be sent to the address below on a Unix tar format tape, either reel-to-reel or 1/4 inch cartridge.

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



If you have an interest in the forthcoming Common Lisp standard, stay tuned. The best thing you can do is to review the draft standard when it comes out; I'll announce it in this column (among other places). If we are to produce a standard that truly reflects the needs of many users and implementors, there must be timely review and feedback. When the time comes, let X3J13 know what you think!

THE WORLD ACCORDING TO SIR BLOOPER

It is not infrequent that one reads an otherwise well-written piece of prose to find abuse of the English language. The would-be author should be ever vigilant to the possibility of abuse. Here is a splendid, albeit juvenile, example of what I mean. Its source is not known to me. I've copied the text as I received it from an anonymous donor.

The World According to Student Bloopers

=====

One of the fringe benefits of being an English or History teacher is receiving the occasional jewel of a student blooper in an essay. I have pasted together the following "history" of the world from certifiably genuine student bloopers collected by teachers throughout the United States, from eighth grade through college level. Read carefully, and you will learn a lot.

The inhabitants of ancient Egypt were called mummies. They lived in the Sarah Dessert and traveled by Camelot. The climate of the Sarah is such that the inhabitants have to live elsewhere, so certain areas of the dessert are cultivated by irritation. The Egyptians built the Pyramids in the shape of a huge triangular cube. The pramids are a range of mountains between France and Spain.

The Bible is full of interesting caricatures. In the first book of the Bible, Guinnesses, Adam and Eve were created from an apple tree. One of their children, Cain, once asked, "Am I my brother's son?" God asked Abraham to sacrifice Isaac on Mount Montezuma. Jacob, son of Isaac, stole his brother's birth mark. Jacob was a patriarch who brought up his twelve sons to be patriarchs, but they did not take to it. One of Jacob's sons, Joseph, gave refuse to the Israelites.

Pharaoh forced the Hebrew slaves to make bread without straw. Moses led them to the Red Sea, where they made unleavened bread, which is bread made without any ingredients. Afterwards, Moses went up on Mount Cyanide to get the ten commandments. David was a Hebrew king skilled at playing the liar. He fought with the Philatelists, a race of people who lived in Biblical times. Solomon, one of David's sons, had 500 wives and 500 porcupines.

Without the Greeks we wouldn't have history. The Greeks invented three kinds of columns-Corinthian, Doric, and Ironic. They also had myths. A myth is a female moth. One myth says that the mother of Achilles dipped him in the River Stynx until he became intollerable. Achilles appeared in "The Iliad", by Homer. Homer also wrote "The Oddity", in which Penelope was the last hardship that Ulysses endured on his journey. Actually, Homer was not written by Homer but by another man of that name.

Socrates was a famous Greek teacher who went around giving people advice. They killed him. Socrates died from an overdose of wedlock.

SIGPLAN can help you...

If you want to involve yourself in standards work, but need financial support, or you are a student who has a paper to present, but have insufficient funds to get to 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 wishing to advance their professional goals. The intention is to provide assistance to members who find that they have no other means to pay for their involvement in various worthwhile activities. A limited amount is budgeted for PAC activities in any one year. Applications are treated on a first-come, first-served basis. PAC expects to fund applications for (in order of priority):

1. annual fees for association with a standards or other voluntary committee
2. registration fees, accomodation and/or travel expenses for student attendance at a workshop or conference where the applicant is to present a paper
3. any other activity deemed appropriate by the committee.

It is *essential* that applicants pay strict attention to the application submission requirements listed below; failure to do so will delay consideration of the application and miss necessary deadlines. Applications should be made reasonably ahead of time (in particular, to facilitate use of inexpensive fares). It is recommended to make applications **at least three months** before the activity is to begin or **two months** for supported attendance at SIGPLAN-sponsored conferences.

Applications must include:

1. a résumé (one page, if possible)
2. a description of the activity and relevance to SIGPLAN
3. a budget
4. a statement of all available and prospective support with an endorsement from the applicant's employer or institution (e.g. department chairman).
5. 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.)

PAC members, listed below, will review applications and subsequently make recommendations to the chair of SIGPLAN. Applications should be submitted in parallel to:

Barbara Ryder
Department of Computer Science
Hill Center, Busch Campus
Rutgers University
New Brunswick, NJ 08903
USA

Julian Padget
Laboratory for Computer Science
University of Bath
Bath, Avon, BA2 7AY
United Kingdom

Mary Lou Soffa
307 MIB
Computer Science Department
University of Pittsburgh
Pittsburgh, PA 15260
USA



Preliminary CALL-FOR-PAPERS

ECOOP-OOPSLA '90

Joint Conference on

Object-Oriented Programming: Systems, Languages, and Applications

21-25 October 1990

The Westin & The Chateau Laurier Hotels, Ottawa, Canada

Chaired by

D. Thomas (Carleton Univ.) and P. Cointe (Rank Xerox France)

Technical papers and survey papers are invited in the topics listed below and in closely related areas. Technical papers should describe original work. Survey papers should discuss existing work in a way that provides new insights.

Languages
Design
Applications
Teaching

User Interfaces
Implementation
Concurrency

Theory
Tools and Environments
Databases

Papers will be selected on the basis of their originality, the significance of their contribution to the state of the art, and the quality of their presentation.

Papers must be written in English and may not exceed 5000 words. A separate cover sheet should indicate the title, authors' names, affiliations, addresses (postal and electronic), telephone numbers, 100-word abstract, numbers of words in the paper, and a list of keywords. Authors must also indicate the topic areas of the paper, choosing from among those listed above. The title and abstract should be carefully written so that they reflect the contents of the submitted paper properly. (Titles and abstracts without author/affiliation MAY be distributed to the program committee members before choosing reviewers.) All accepted papers will be published in the conference proceedings, and authors will be required to sign an ACM copyright release form. Six copies of submitted papers accompanied with cover sheets must be received by March 1, 1990. In addition, authors are asked to send electronic forms of separate cover sheets (NOT whole papers). Authors will be notified of acceptance by May 18, 1990 and the final camera-ready versions of accepted papers will be due on June 30, 1990. Late papers, papers that exceed 5000 words, and papers that are not accompanied with the requested cover sheet information will not be reviewed. Please send physical copies of papers with cover sheets to:

Akinori Yonezawa, Ecoop/Oopsla90 Program Chair
Dept. of Information Science, Faculty of Science
University of Tokyo, 7-3-1 Hongo, Bunkyo-ku
Tokyo, 113 Japan

Please send electronic copies of cover sheets to:

oopsla90@u-tokyo.ac.jp

Program Committee includes:

A. Yonezawa, Chair	(Univ. of Tokyo)	K. Beck	(Apple)
A. Borning	(Univ. of Washington)	L. Cardelli	(DEC)
S. Cook	(Queen Mary College)	J.-M. Hullot	(NeXT)
K. Kahn	(Xerox)	W. Lalonde	(Carleton Univ.)
H. Lieberman	(MIT and Univ. of Paris)	O. Madsen	(Aarhus Univ.)
A. Snyder	(HP)	L. Steels	(Brussel Free Univ.)
B. Stroustrup	(AT&T)	D. Tschritzis	(Univ. of Geneva)
M. Tokoro	(Keio Univ.)	J. White	(Lucid)

LISP AND SYMBOLIC COMPUTATION: An International Journal

Scope:

- Programming language notations for symbolic computing (e.g., object-oriented programming, data abstraction, parallelism, lazy evaluation, infinite data objects, self-reference, message-passing, generic functions, inheritance, encapsulation, protection, metaobjects).
- Implementations and techniques (e.g., specialized architectures, compiler design, combinatory models, garbage collection, storage management, performance analysis, smalltalks, flavors, common loops, etc.).
- Programming logics (e.g., semantics and reasoning about programs, types and type inference).
- Programming environments and tools (e.g., knowledge-based programming tools, program transformations, specifications, debugging tools).
- Applications and experience with symbolic computing (e.g., real-time programming, artificial intelligence tools, experience with LISP, object-oriented programming, window systems, user interfaces, operating systems, parallel/distributed computing).

Editors-in-Chief:

Richard P. Gabriel, Lucid, Inc. and Guy L. Steele Jr., Thinking Machines, Inc.

Articles of Volume 3:3, September 1990 include:

A Preprocessor Based Implementation of Common Lisp, John R. Diamant, Gerald P. Duggan; *A New Approach to Procedures with Variable Arity*, R. Kent Dybvig, Robert Hieb; *LogScheme: Integrating Logic Programming into Scheme*, Erik Ruff, Daniel Weise; *Combinator Evaluation of Functional Programs With Logical Variables*, Göran Båge, Gary Lindstrom.

Submissions and more information contact:

Jan Zubkoff
Associate Editor, LASC
Lucid, Inc.
707 Laurel Street
Menlo Park, CA 94025
jlz@lucid.com
415/329-8400

For subscriptions please contact:

Kluwer Academic Publishers
P.O. Box 358
Accord Station
Hingham, MA 02018-9990
617/871-6600