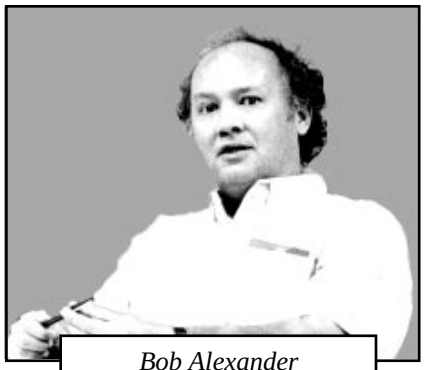
Second Workshop on the Icon Programming Language

July 25-27, 1990
Flagstaff, Arizona

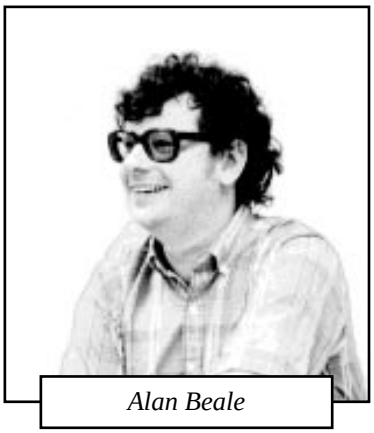
Ralph E. Griswold and Madge T. Griswold

The University of Arizona and The Bright Forest Company

In the summer of 1988, small group of persons closely associated with Icon met in an informal working environment in Flagstaff, Arizona [1]. A second workshop was held on July 25-28, 1990. Thirty-one persons were invited to the second workshop. Of these, twenty-one attended. The site, on the campus of Northern Arizona University, was the same as for the first workshop, with Steve Wampler as host.



Bob Alexander



Alan Beale

The attendees met on the evening of July 25 for introductions and discussion of the agenda. Full-day meetings were held on July 26 and 27. The structure of the workshop was flexible and the atmosphere was informal; rather then formal presentations, most of the time was devoted to discussion.

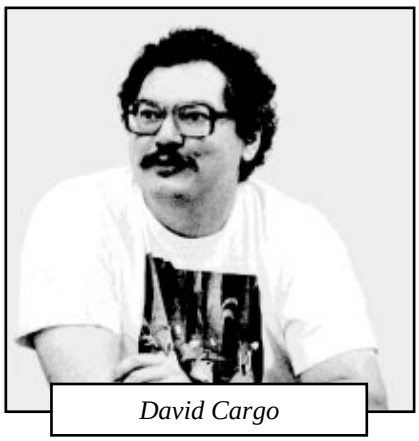
Thursday Morning, July 26

Ralph Griswold opened the first session with a brief history of Icon and a report on its present status. He pointed out that work on Icon started over twelve years ago and that the eighth version of Icon was released in May, along with the second edition of the Icon book.

Peter Bigot

He noted that Icon has been supported mostly by research grants from the National Science Foundation and that Icon itself is a by-product of this research program, not its main focus. He also explained the current prospects for continued grand funding and the need for new research directions.

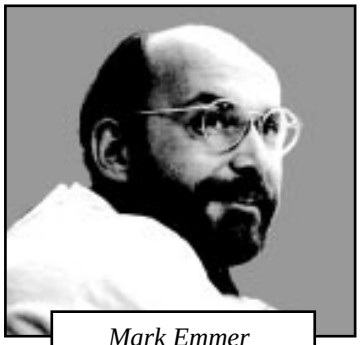
Ralph concluded by commenting that support for Version 8 of Icon would be continued, and that there were a few loose ends to be taken care of, but that there were no plans for another version.



David Cargo

In the discussion that followed, Bob Goldberg expressed concerns about the end of Icon language innovation. David Cargo asked about the possibility of future support from a major computer vendor. Ralph Griswold

commented that computer vendors don't undertake the support of software unless they have to.

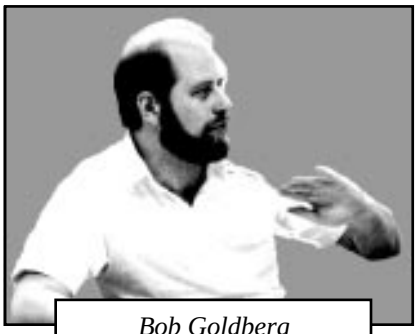

Mark Emmer

Next the attendees described their uses of Icon. These ranged from large applications for constructing databases to "throw-away" programs to perform simple "one-shot" jobs. The word "utility" was mentioned frequently, as were references to using Icon as a replacement for C, Awk, and other programming languages. There were numerous interesting anecdotes about ways that Icon had made something possible that otherwise would not have been done. Joe Hall described his (rather painful) experience with moving a large Icon application from UNIX to MS-DOS.

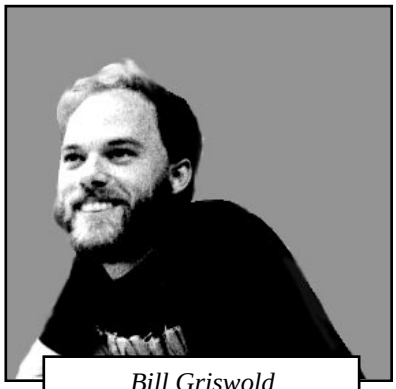

Mary Fernandez

The second morning session focused on features of Icon that persons wanted added or changed. David Cargo presented his "wish list" with modularization to support development of large programs and more flexible input and output conversion as his main needs.

There was considerable discussion of the need for an Icon preprocessor to provide file inclusion, manifest constants, and macro definitions. Members of the group had different opinions about whether i/o conversion should be treated as a general feature of Icon, by system-specific extensions, or coded in Icon itself.


Bob Goldberg

Steve Stone mentioned possible extensions to Icon for business applications, such as an SQL server. And inevitably, the question of a windows interface was discussed with, again, a diversity of opinion as to whether Icon should develop its own interface, adopt an existing one, or have system-specific alternatives.


Bill Griswold

Joe Hall remarked that if Icon is frozen, they could now make customized modifications knowing future research isn't going to change the language.

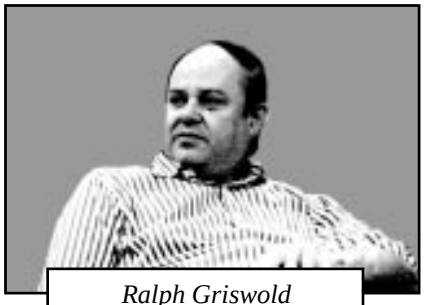
Thursday Afternoon, July 26


Madge Griswold

The first afternoon session on Thursday was devoted to object-oriented versions of Icon. Clint Jeffery briefly described Idol [2], an object-oriented version of Icon built on top of Icon and implemented by a preprocessor written in Idol that translates that program to Icon programs.

There was discussion of motivation for object-oriented features and the pros and cons of the "extra layer" involved in Idol. Most persons viewed object-oriented features as being primarily useful for structuring large programs.

The next afternoon session was devoted to new work at The University of Arizona on a visual programming environment for Icon. Ralph Griswold described the overall goals of the project — to develop an integrated set of tools for writing, debugging, and maintaining Icon programs with an emphasis on visualization. He also commented that


Ralph Griswold

while a debugger would likely be a component of such an environment, he preferred a larger view of program development in which information about program performance and behavior were important components.


Joe Hall

He mentioned Icon's memory monitor [3] as an example of an existing tool in this spirit. He demonstrated the memory monitor using an application written for the Macintosh by Mark Emmer.

Clint Jeffery described an X-Windows interface for Icon [4] and his work on developing a version of Icon that would allow several programs to operate under the same interpreter [5]. Such programs could communicate, and it even would be possible to write tools such as debuggers in Icon itself.


Clint Jeffery

Clint described an Icon code browser he's written [6] and Nick Kline described a tool he has written to visualize procedure activity in Icon.

The afternoon ended with a discussion of teaching Icon. Ralph Griswold, Bob Goldberg, Joe Hall, David Cargo, and Steve Wampler discussed their experiences. It was generally acknowledged that it would be helpful to have teaching materials available for others to use.


Nick Kline

Steve Stone described the position of Icon as a second language and the difficulty programmers had "changing gears".

Friday Morning, July 27

Friday morning started with a presentation by Ken Walker of the Icon compiler he has written [7]. He described the organization of the compiler, which includes a runtime system written in an extension to C, the library and the database it produces, and the compiler proper, which generates C code.

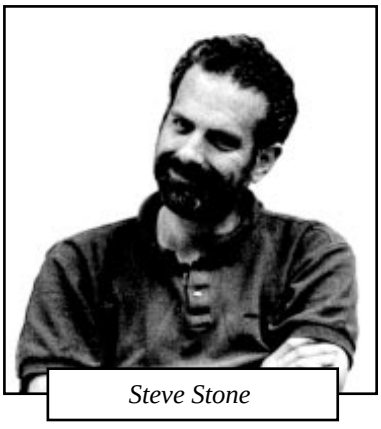

Bill Mitchell

The main subjects addressed by the compiler are type inference, allocation of space for temporary results, and the optimization of expression evaluation. The compiler presently is running, although not all optimizations are incorporated.


Jerry Nowlin

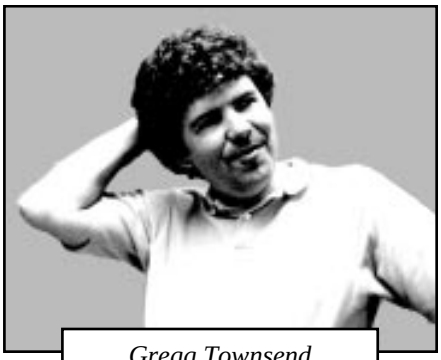
Ken presented figures comparing the performance of the present compiler to the existing interpretive implementation of Icon.

There was discussion of the features of Icon supported and not supported by the compiler, the options it provides, and how it might be used and extended.


Steve Stone

The morning session concluded with a description of dynamic hashing for sets and tables by Gregg Townsend. There was extensive discussion about the possible consequences of adding or deleting elements from a set or table while elements

during generation. Steve Stone observed that dynamic hashing for Icon was the consequence of a published algorithm and asked how much cross fertilization there was between theory and practice in Icon.



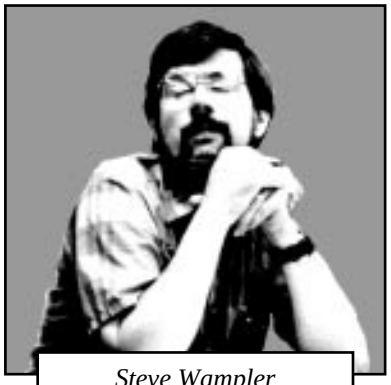
Gregg Townsend

Friday Afternoon, July 27

The afternoon session started with a presentation by Mary Fernandez of her work on alternatives to Icon's mark-and-compact form of garbage collection. She presented performance figures of different strategies she has implemented. It appears that the most significant improvements can be obtained in situations where Icon has really large storage regions. Problems with systems such as MS-DOS that have small regions also were discussed, and Cheyenne Wills mentioned his plans for multiple storage regions.

Ken Walker

The workshop ended with a session on interfacing Icon to the "real world". Mark Emmer listed the facilities needed in the language and implementation to support the Macintosh visual environment, including operating-system calls, memory allocation, raw data access, call back, record structures, and preprocessor facilities.



Steve Wampler

Bob Alexander described how he did much of this on the Macintosh under MPW. There was discussion of the level at which such features should be provided and in what sense they would or should be portable. Alan Beale appealed for a solution that would not leave IBM 370 systems isolated from all others. Many suggestions were made, but there was no consensus.

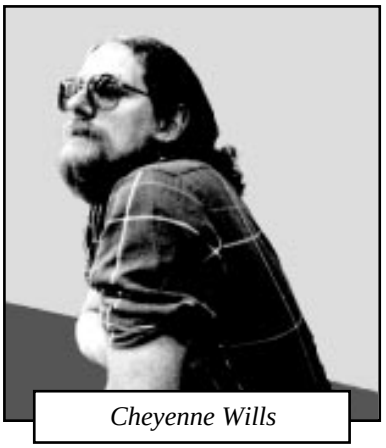
Beth Weiss

The workshop ended on an up-beat note with the expectation of a third workshop two years hence.

References

1. "A Workshop on the Icon Programming Language", Ralph E. Griswold, Icon Project Document IPD61, Department of Computer Science, The University of Arizona, 1988.

2. "Programming in Idol: An Object Primer", Clinton L. Jeffery, Technical Report TR 90-10, Department of Computer Science, The University of Arizona, 1990.



Cheyenne Wills

3. "The Visualization of Dynamic Memory Management in the Icon Progamming Language", Ralph E. Griswold and Gregg M., Townsend, Technical Report TR 89-30, Department of Computer Science, The University of Arizona, 1989.

4. "The Icon Windows Interface", Clinton L. Jeffery, draft report, Department of Computer Science, The University of Arizona, 1990.

5. "The Icon Programming Environment", Clinton L. Jeffery, draft report, Department of Computer Science, The University of Arizona, 1990.

6. “Ibrowse — An Icon Browser”, Clinton L. Jeffery, draft report, Department of Computer Science, The University of Arizona, 1990.

7. “A Compiler for Icon”, in *Icon Newsletter 33*, Department of Computer Science, The University of Arizona, 1990.

Attendees

Robert J. Alexander
3514 La Terrace Circle
San Jose, CA 95123

alex@metaphor.com

Alan Beale
Senior Systems Scientist
C Compiler Development
SAS Institute Inc.
1 SAS Circle
Cary, NC 27512-8000

sas!sasmail!sdcmvs!sasarb@mcnc.org

Peter A. Bigot
Department of Computer Science
The University of Arizona
Tucson, AZ 85721

pab@cs.arizona.edu

David S. Cargo
Cary Research, Inc.
655F Lone Oak Drive
Minneapolis, MN 55121

cargo@cray.com

Mark B. Emmer
Catspaw, Inc.
P.O. Box 1123
Salida, CO 81201

emmer@cs.arizona.edu

Mary F. Fernandez
Department of Computer Science
Princeton University
35 Olden Road
Princeton, NJ 08544-2087

mmf@cs.princeton.edu

Robert E. Goldberg
4401 West Estes
Lincolnwood, IL 60646

goldberg@iitmax.iit.edu

Madge T. Griswold
The Bright Forest Company
5302 East Fourth Street
Tucson, AZ 85711

madge@cs.arizona.edu

Ralph E. Griswold
Department of Computer Science
The University of Arizona
Tucson, AZ 85721

ralph@cs.arizona.edu

William G. Griswold
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195

wgg@cs.washington.edu

Joe T. Hall
AT&T Bell Laboratories
Warrenvilee & Naperville Roads
Naperville, IL 60566

att!ihcup!pax

Clinton L. Jeffery
Department of Computer Science
The University of Arizona
Tucson, AZ 85721

cjeffery@cs.arizona.edu

Nick Kline
Department of Computer Science
The University of Arizona
Tucson, AZ 85721

kline@cs.arizona.edu

William H. Mitchell
Sunquest Information Systems
930 N. Finance Center Drive
Tucson, AZ 85710

whm@sunquest.com

Jerry Nowlin
AT&T Bell Laboratories
IN-12-226
1100 E. Warrenville Road
Naperville, IL 60566

att!iwtqg!nowlin

Steve Stone
S. B. Stone & Company
5000 Rockside Road, Suite 500
Cleveland, OH 44131

Gregg M. Townsend
Department of Computer Science
The University of Arizona
Tucson, AZ 85721

gmt@cs.arizona.edu

Kenneth Walker
Department of Computer Science
The University of Arizona
Tucson, AZ 85721

kwalker@cs.arizona.edu

Steve Wampler
College of Engineering
Northern Arizona University
Box 15600
Flagstaff, AZ 86011

sbw@turing.cse.nau.edu

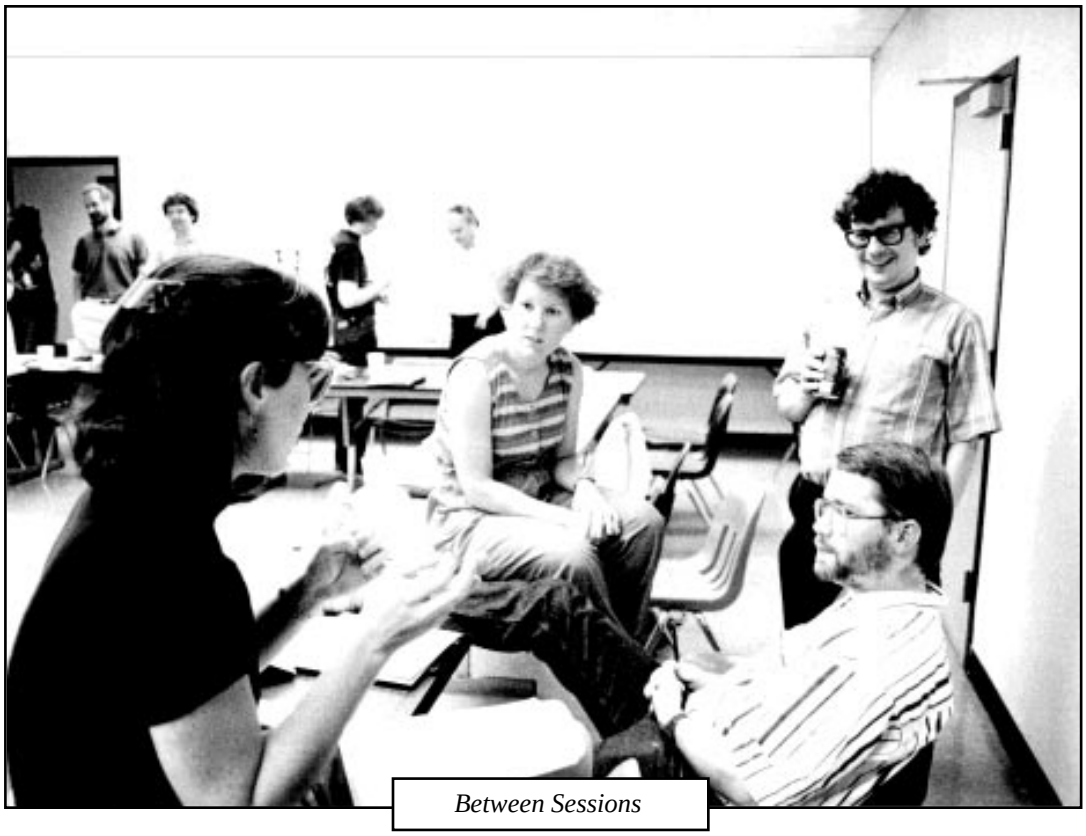
Beth Weiss
Department of Computer Science
The University of Arizona
Tucson, AZ 85721

bweiss@cs.arizona.edu

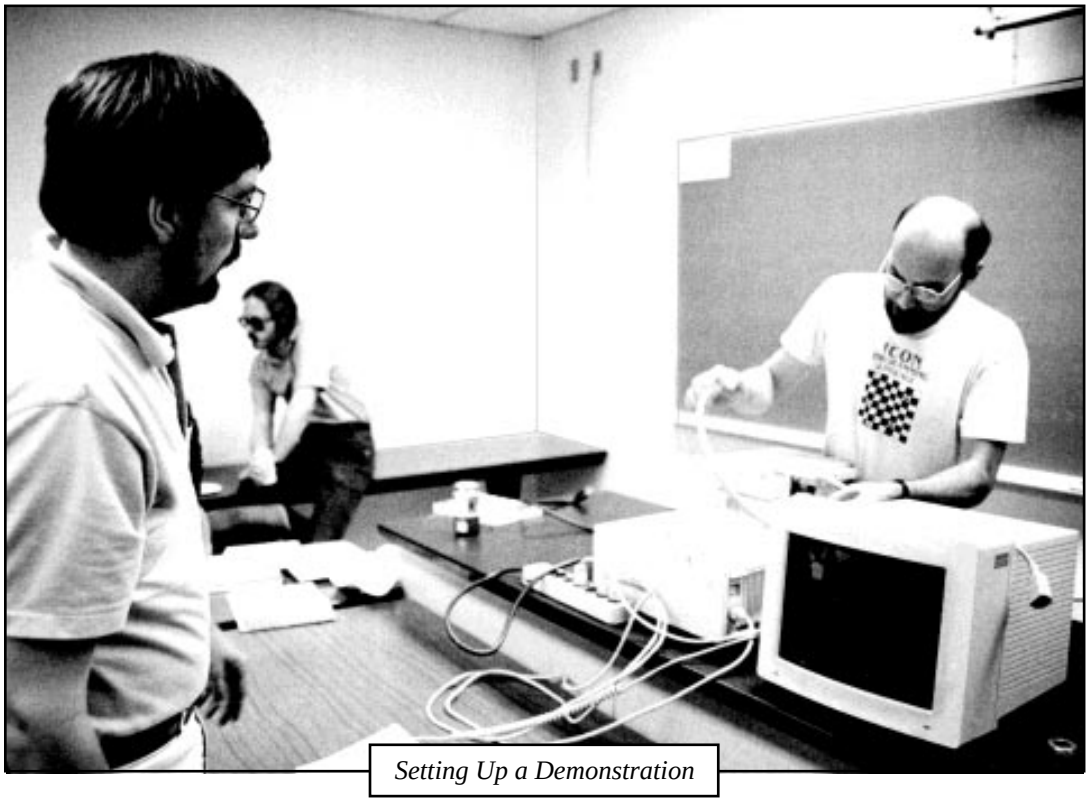
Cheyenne Wills
12 West Locust Street
Mechanicsburg, PA 17055

cheyenne@cs.arizona.edu

In Session



Between Sessions



Setting Up a Demonstration

Group “Portrait”