Third Workshop on the Icon Programming Language

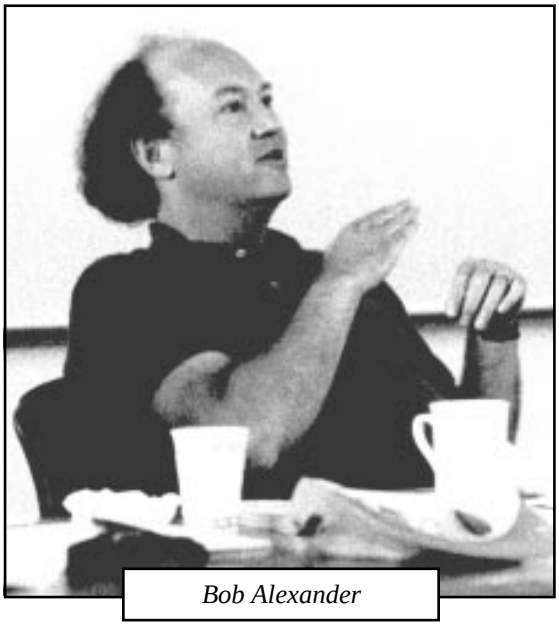
September 10-11, 1992
La Jolla, California

Ralph E. Griswold and Madge T. Griswold
The University of Arizona and The Bright Forest Company

Icon workshops provide an opportunity for persons with an involvement in Icon to meet and discuss their mutual interests.

The first two workshops were held in Flagstaff, Arizona in 1988 and 1990. The third workshop was held on September 10-11, 1992, in La Jolla, California. The workshop was sponsored by the Department of Computer Science and Engineering of the University of California, San Diego. Bill Griswold served as local host.

Fifteen persons attended this workshop. Those who arrived early met on the evening



Bob Alexander

of September 9 to discuss the agenda. Full-day meetings took place on September 10 and 11. The workshop atmosphere was informal, with a mixture of presentations and discussion.

Thursday Morning, September 10

Bill Griswold opened the first session with a welcome, after which the persons attending introduced themselves and described their interests.

Alan Beale

Ralph Griswold followed with a report on the state of the Icon Project. He summarized the situation with respect to Icon as it existed two years ago, at the time of the last workshop: Version 8.0 of Icon had been released; X-Icon, a multi-thread version of Icon (MT Icon), and the optimizing compiler were in progress; program visualization tools were planned and a visual programming environment was proposed.

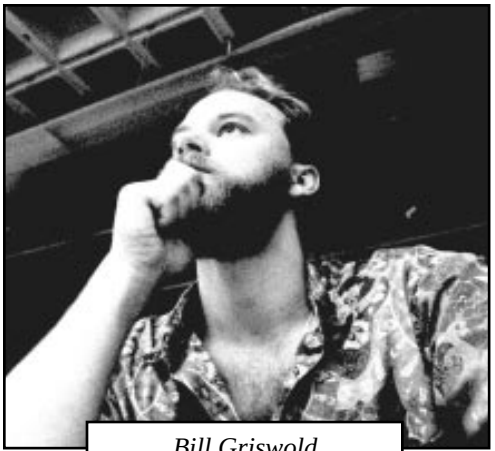
Next he described the present situation: the compiler and interpreter run-time systems have been combined and released, along with X-Icon, for UNIX and VMS; MT Icon and event-monitoring instrumentation is working; an X-Icon toolkit and interface builder are working; X-Icon for OS/2 has been devel-

David Cargo

oped; a framework for program visualization is in place and some visualization tools have been written.

He then described the status of the implementation for various platforms and the present state of distribution. He listed the goals of the Icon Project as bringing the implementation of Icon for all platforms up to date, releasing the compiler for 32-bit MS-DOS platforms, releasing MT Icon and the X-Icon toolkit and interface builder, and continuing work on program visualization.

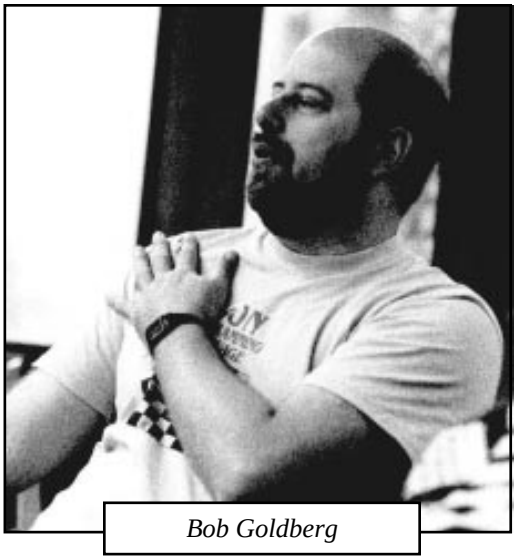
Among other possibilities, he mentioned implementations of X-Icon for other platforms, the extension of X-Icon facilities, refinements to the compiler, and the development of a visual programming environment for Icon.



Bill Griswold

Discussion following the presentation covered the grant funding situation, the general problem with financial support for Icon, and future prospects.

Next Ken Walker described the new optimizing compiler for Icon. Among goals, he listed optimizations to eliminate unnecessary run-time type checking, placing operations in line, static allocation of temporary variables, portability, language extensibility, support for the complete Icon language, and a production-quality implementation.



Bob Goldberg

He described how the compiler is organized, its database of run-time information, how this information is obtained from code for the run-time system, and the processes of building the compiler and running it. He then gave figures for the resources required to run the compiler and provided timings comparing the compiler to the interpreter. Ken also showed how Icon programs could be cross compiled, and what was involved in building the interpreter with the new run-time system. He concluded with a status report, describing what the compiler can do and on what platforms it presently is implemented.

The morning session ended with a presentation on X-Icon by Gregg Townsend. He gave an overview of X-Icon, its Xlib interface, how windows are created and managed, display functions, attributes associated with windows, graphic contexts, input functions, and event keywords.

Madge Griswold

Next he described X-Icon extensions to the basic X model: retained windows, terminal emulation, and the color model. He listed features of X that are not supported by X-Icon, including subwindows, image synthesis, and numerous small matters. Among X-Icon applications to date, he listed visualization tools, an interface builder, an editor, and several applications done as projects in a string and list processing class.

Clint Jeffery

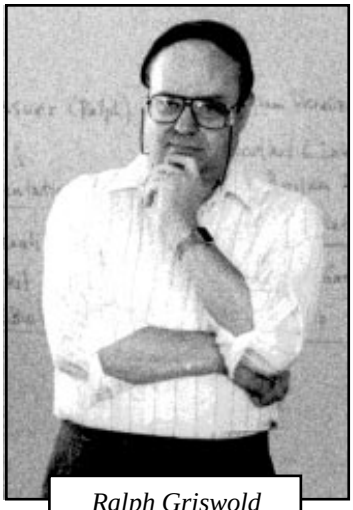
He noted that X-Icon is included in Version 8.7 of Icon for UNIX and VMS and that an OS/2 port is in progress. Gregg closed with ideas for future extensions.

Thursday Afternoon, September 10

The Thursday afternoon session began with a demonstration of X-Icon programs by Gregg Townsend and Bob Alexander.

Ken then continued his presentation of the Icon compiler with a description of RTL, the language in which the Icon run-time system is now written. He explained that the motivation for RTL is to provide the information the compiler needs about run-time operations. RTL, an extension of C, is designed so that this information can be extracted automatically while RTL is being translated into C code that is used to build the run-time system.

Ken described the mechanisms RTL provides for expressing Icon type conversion and abstract type computations. He showed examples of writing a built-in function and an operator in RTL. He then described the type specification system, which allows new types to be added to Icon.



Ralph Griswold

Following his talk on RTL, Ken gave an overview of ISIcon: its new built-in functions, module-level scoping, function tracing, and enhanced diagnostics. He also described variants of ISIcon: ISIcon/SI, which has character windowing features for UNIX/terminfo and ISIcon/VOICE, which has features for voice processing.

After a mid-afternoon break, Kelvin Nilsen described his plans for an Icon dialect for instruction, based on Kamin's text for an interpreter-based approach to teaching programming languages. The text presents programming language concepts by simplified dialects cast in Lisp syntax. Students

write programs in each language. Kelvin explained how the use of a common simplified syntax enables students to experiment with different features and compare different languages without having to learn their syntaxes.

Kelvin listed the features of Icon that might be emphasized in such a context and what the objectives should be. There was discussion about what should and should not be included in such a subset of Icon and how effective such an approach might be.

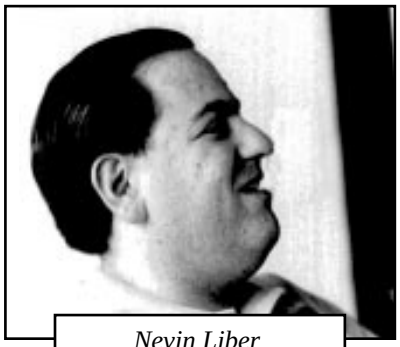
The afternoon session ended with a brief description of MT Icon by Clint Jeffery. He described the multi-thread model of execution, its general characteristics, how multiple programs are loaded and executed under a single invocation of the Icon interpreter, and what can be done with MT Icon.

Kelvin Nilsen

Friday Morning, September 11

The second day of the workshop started with a general discussion of language issues. Bob Goldberg raised the issue of adding a preprocessor to Icon, even if it only had the capability for including files and defining constants. Ralph pointed out that there was a stand-alone preprocessor in the Icon program library. This led to discussion of the library, concern that it was generally under-utilized, and what could be done to make it more accessible.

David Cargo then asked about the motivation of the new sortf() function. A discussion of sorting followed, with Bob Alexander pointing out that there was a generalized sorting procedure in the program library.



Nevin Liber

Steve Stone asked if "catch" and "throw" could be done in Icon and Steve Wampler said that additional mechanisms would needed for these.

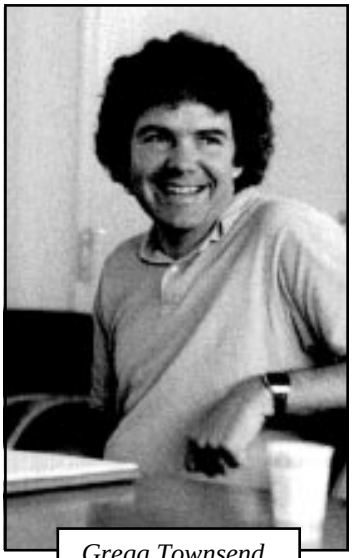
Bob Alexander next presented a number of suggestions for new language features, including a way of getting a record field number from its name, new sorting features, an encapsulation mechanism similar to ISIcon's modules, as well as several other ideas.

Following the discussion, Clint made a presentation on the instrumentation and monitoring facilities that had been added to MT Icon. He noted that the term multi-thread was inaccurate, as Steve Stone had discussed with him and that a better term was needed. He explained the motivation for producing information about program execution that could be used by monitors: debugging, performance tuning, education, and generally getting a better understanding of program behavior. With the features

Jon Pearkins

added to MT Icon, monitors can be written in Icon itself, and they can get direct access to the program being monitored. He mentioned the problems with monitoring and described approaches to their solution.

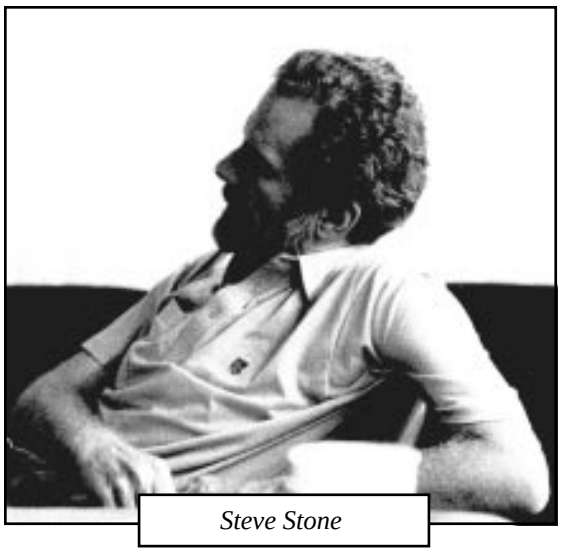
Following a mid-morning break, Ralph described work that had been done in program visualization using execution monitors that present their results in the form of images. He mentioned the earlier memory-monitoring system written by Gregg, more recent use of visual metaphors for displaying storage allocation information, fish-eye views and other techniques for presenting information in a small amount of screen space, and monitors that can provide different views of the same program activity. He concluded by describing Eve, an execution monitor coordinator written by Clint to allow several monitors to operate on the same program at the same time.



Gregg Townsend

Ralph then described the X-Icon tool kit developed by Jon Lipp to supplement the low-level Xlib functionality of X-Icon with Icon procedures to provide "vidgets" (virtual input/output devices) such as buttons, toggles, check boxes, menus, sliders, and scrollbars. He showed examples of programs that illustrate the use of vidgets.

Next he described Mary Cameron's X-Icon interface builder that provides an interactive, direct-manipulation system for building X-Icon applications that use vidgets.



Steve Stone

Friday Afternoon, September 11

Following lunch, Ralph demonstrated some of the program visualization monitors and the X-Icon interface builder.

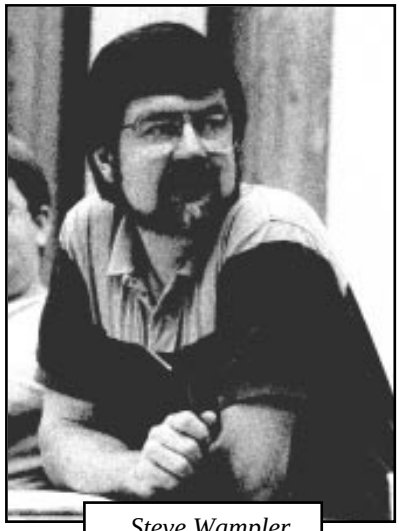
After the demonstration, there was a general discussion of implementation issues. Storage allocation was discussed and Clint described the techniques used in managing multiple storage regions. Bob Alexander asked about the possibility of dynamically linked functions. Steve Stone asked about the possibility of a feature for saving large structures for use in a later program execution. Bob Alexander suggested improvements to Icon's tracing facilities to limit tracing to selected procedures.

Steve Stone asked about the purpose of the discussion of language and implementation issues when the Icon Project appeared not to have the resources to make significant changes to Icon. Ralph responded that such discussions were interesting in their own right, that something might, in fact, come out of them, and persons outside the Icon Project could make changes also.

The possibility of revising the Icon implementation book was raised. Madge and Ralph described some of the difficulties they

had experienced with publishers and why such a revision was unlikely.

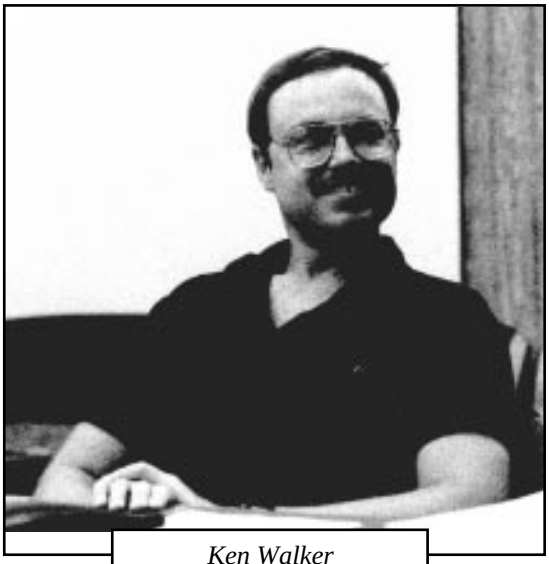
Following a mid-afternoon break, Kelvin presented a summary of some of his work on real-time garbage collection. He pointed out that much of the programming effort on large programming projects often was directly related to memory management. He mentioned that the costs of DRAM are low enough that it can be used in place of virtual memory and that VLSI components represent only a small fraction of the cost of a complete system. These factors motivate hardware support for memory management.



Steve Wampler

He said profiling shows garbage collection requires up to half of the CPU resources for some applications, but profiling does not reveal all the costs, such as tagging and tending pointers.

He described an architecture that provides hardware support for garbage collection and presented code-generation models for C++ to use this support. Kelvin then showed several graphs illustrating the improvements in performance that could be achieved with hardware support for garbage collection and ended with comments on the potential implications for Icon.



Ken Walker

The workshop ended with a general discussion. Ralph mentioned the possibility of a programming environment for Icon and extensions to the X-Icon facilities. Clint mentioned the possibility of using MT Icon for modularizing large programs.

Bob Goldberg asked Ralph if he could visualize a successor to Icon and what features would it have. Ralph responded that it would have fewer features with more expressiveness and better unification of the features it did have. He suggested that such a language might benefit from a more object-oriented basis.

Demonstrating X-Icon

The workshop ended with thanks and compliments to Bill for the excellent conference arrangements.

Participants:

Robert J. Alexander
6603 Bose Lane
San Jose, CA 95120
alex@metaphor.com

Alan Beale
3750 Jamestown Circle
Raleigh, NC 27609
sasarb@mvs.sas.com

David S. Cargo
1735 Rome Ave.
St. Paul, MN 55116
cargo@cray.com

Robert E. Goldberg
4401 West Estes
Lincolnwood, IL 60646
goldberg@iitmax.iit.edu

Madge T. Griswold
The Bright Forest Company
5302 E. 4th Street
Tucson, AZ 85711-2304
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
UC San Diego
Computer Science
and Engineering, 0114
La Jolla, CA 92093-0114
wgg@cs.ucsd.edu

Clinton L. Jeffery
1119 East 12th Street, #4
Tucson, AZ 85719
cjeffery@cs.arizona.edu

Nevin J. Liber
243 Buena Vista Avenue
No. 803
Sunnyvale, CA 94086-4869
nevin@apple.com

Kelvin Nilsen
Department of Computer Science
Iowa State University
226 Atanasoff Hall
Ames, IA 50011-1040
kelvin@cs.iastate.edu

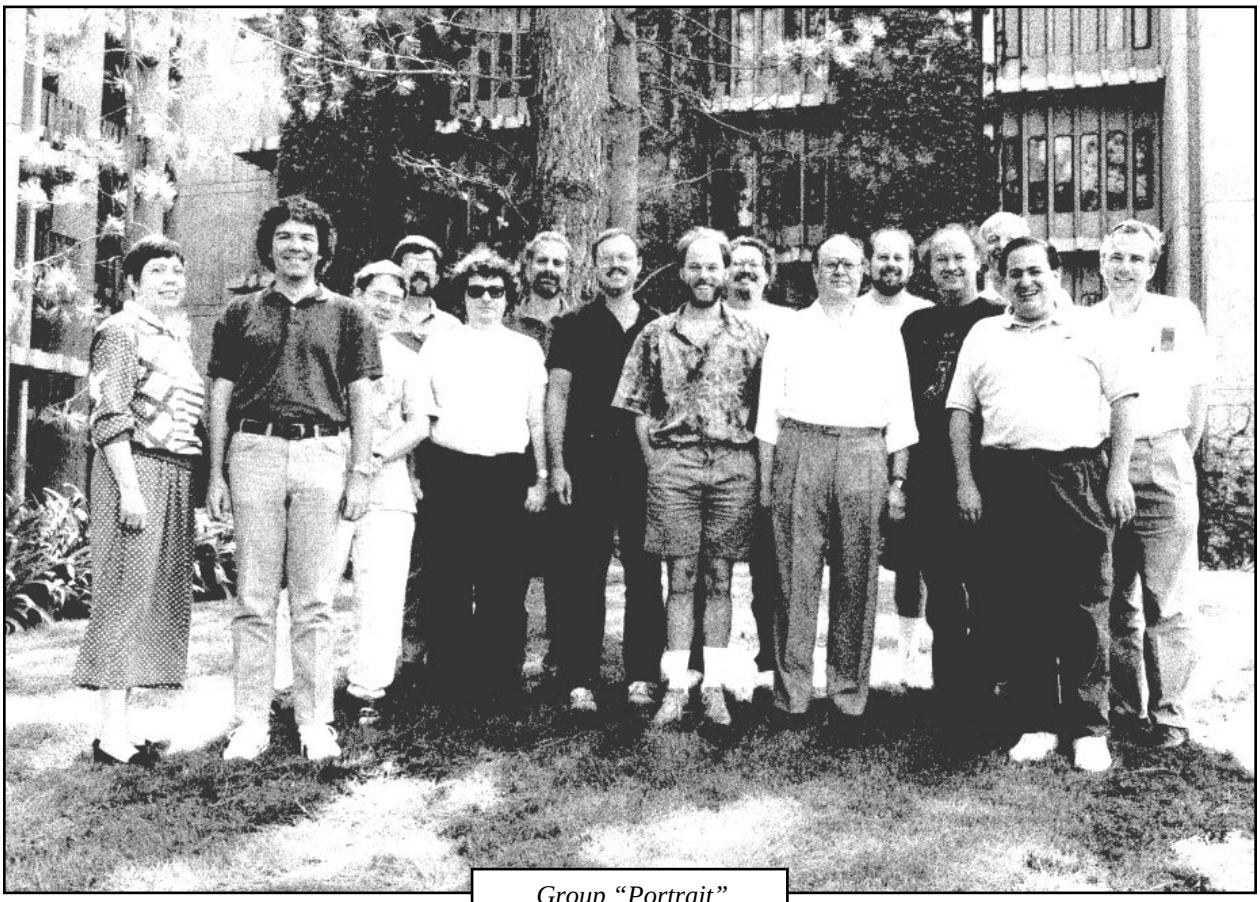
Jon Pearkins
Certified Software Specialists Ltd.
54015 Range Road 212
Ardrossan, Alberta T0B 0E0
CANADA
71231.3005@compuserve.com

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

Gregg Townsend
Department of Computer Science
The University of Arizona
Tucson, AZ 85721
gmt@cs.arizona.edu

Kenneth Walker
415 E. University #310
Tucson, AZ 85705
kwalker@cs.arizona.edu

Stephen B. Wampler
1409 West Louise Way
Flagstaff, AZ 86001
sbw@turing.cse.nau.edu



Group "Portrait"