THE UNIVERSITY OF ARIZONA
TUCSON, ARIZONA 85721

DEPARTMENT OF COMPUTER SCIENCE

Icon Newsletter #5

Ralph E. Griswold

December 31, 1980

1. Version 2 of Icon

1.1 Status

As mentioned in the last Newsletter, Version 2 of Icon is stable and no changes to it are anticipated. The portable system is still available from us (see the form at the end of this Newsletter) and implementations for a variety of computers are completed or in progress.

1.2 Existing Implementations

The DEC-10 and CDC 6000/Cyber implementations are still available from us (see the forms at the end of this Newsletter). Other available implementations are for the IBM 370 (and compatible computers), the VAX-11/780, and the Cray-1.

The IBM 370 implementation (which runs under OS/360) and the VAX-11/780 implementation (which runs under VMS) are available from

William H. Mitchell
University Systems Analysis and Control Center (USACC)
Box 50298
338 Daniels Hall
North Carolina State University
Raleigh, North Carolina 27650

Send a 2400′ magnetic tape and $10.00 for handling and postage (for $20.00 they will supply the tape).

The Cray-1 implementation (which operates under COS) will be supplied on receipt of a 2400′ tape by

Bob Cave
Institute for Defense Analyses
Thanet Road
Princeton, New Jersey 08540

An implementation for the ICL 2900 is nearing completion. For information contact

Michael J. Garside, Deputy Director
Computing Laboratory
Cornwallis Building
University of Kent at Canterbury
Kent, England CT2 7NF

1.3 Corrections to the Portable Implementation

Several errors have been discovered in the portable implementation. Earlier this year a list of these errors and corrections to the source code were distributed to all persons who are known to have Version 2 systems (the corrections apply not only to the portable system itself, but also to implementations for specific computers based on the portable systems). If you have a Version 2 system without these updates, ask us for the list of corrections (see the document request form at the end of this Newsletter).

2. Version 3 of Icon

As mentioned in the last Newsletter, our current work is based on the C implementation of Icon running under UNIX*. The current version is 3.2, which corrects a number of errors in earlier versions. This system is available from us (see the form at the end of this document). A list of corrections, changes, and known bugs related to Version 3.2 accompanies the system, but is also available separately — see the document request form.

A few more errors in the Version 3 manual (TR 80-2) have been discovered. Copies of a new corrigenda are available — see the document request form.

3. Current Research

3.1 Generators and Control Structures

Recently we have been studying the evaluation of expressions in Icon in some detail. The goal of this work is to provide a coherent model to serve as a basis for understanding generators, goal-directed evaluation, and control structures. Not surprisingly, insight gained from this work has led to some new programming idioms and has also suggested changes for future versions of Icon. The results appear in a technical report TR 80-21 (see the document request form).

A better understanding of generators, both in their potential advantages and limitations, has led to the development of new control structures and the concept of *co-expressions*, which bear the same relationship to expressions that coroutines do to procedures. The main advantage of co-expressions is that they liberate generators from their lexical site in program text and allow them to be activated at any time or place in the program. The new control structures and co-expressions are implemented in an experimental extension to Version 3.

3.2 Generators in C

Generators and goal-directed evaluation have proved very useful in allowing concise and natural formulations of many problem solutions. These aspects of Icon are, however, largely independent of many of the other aspects of the language (such as automatic storage management, automatic type coercion, the multiplicity of data types, and string scanning).

This suggests that generators and goal-directed evaluation might be incorporated to advantage in other programming language — specifically in conventional Algol-like languages.

To test this hypothesis, generators and control structures to facilitate their use have been added to C. The resulting language, called Ccon, is implemented by a preprocessor and a few support routines.

The results so far are encouraging. In several cases, it has been possible to write Ccon programs that are considerably more compact and clearer than corresponding C programs. Not surprisingly, the initial results have suggested new control structures and contexts in which the host language influences specific aspects of generators.

---

*UNIX is a trademark of Bell Laboratories.

3.3 Pattern Matching in Icon

One of the original motivations for Icon was the discovery of a way to integrate high-level string processing with conventional processing and hence to avoid the "linguistic schism" formed in languages like SNOBOL4 (see "Alternatives to the Use of Patterns in String Processing", *TOPLAS*, April 1980).

Nevertheless, patterns clearly provide a valuable paradigm for formulating solutions to many string analysis problems. The evidence for this lies not only in the popularity of SNOBOL4, but also in the attempts of Icon programmers to apply string scanning in a pattern-matching fashion.

By observing a few simple coding protocols, pattern matching can be performed in Icon. Indeed, almost all of the pattern-matching mechanism of SNOBOL4 can be written in terms of Icon procedures. This not only demonstrates the capabilities of Icon but also provides a formal description of pattern matching in SNOBOL4. Furthermore, there are many generalizations and extensions, including pattern-directed string transformations and list pattern matching. The results of this work appear in technical report TR 80-25 (see the document request form).

4. Other Icon Documents

In addition to the documents mentioned above, two others are available now:

- "The Use of Character Sets and Character Mappings in Icon", from the May 1980 issue of *Computer Journal*, describes a number of programming techniques using the map function and character sets.
- Icon Observed Coding Laws and Standard Techniques (IconOCLAST) describes a standard style for formatting Icon programs.

5. Programming Corner

The programming corner has proved to be a popular feature of earlier Newsletters. This programming corner comes in the form of puzzles and posed questions, with solutions and answers to appear in the next Newsletter. Asterisks indicate material for Version 3 only.

1. What is the output produced by each of the following expressions?

```
every write((0 | 0) to 7)
every write(0 to 3,0 to 7,0 to 7)
every write(1 | 2 to 3 | 4 by 1 | 2)
every 1 to 3 do every write(1 to 3)
```

2. For arbitrary procedures f(x,y) and g(x,y), what is the sequence of calls produced by

```
every (f | g)(1 to 3, 4 | 5)
```

3.* Given

```
s1 := "aeiou"
s2 := "abecaeioud"
```

what are the outcomes of

```
(find | upto)(s1,s2)
(find | upto)(s2,s1)
(if size(s1) > size(s2) then upto else find)(s1,s2)
```

4. What are the outcomes of the following expressions? (Note any that produce errors.)

```
(x | y) := 3
(x & y) := 3
(1 & x) := 3
(x & 1) := 3
(x + 1) := 3
```

5. Given the procedure

```
procedure drive(x)
  fail
end
```

What is the output produced by

```
drive(write(1 to 7))
drive(write(0 to 7,0 to 7))
```

6.* What does execution of the following program do?

```
procedure main()
   f(f := write,f)
end
```

7. The following procedure is proposed as a generator of "words" — strings of consecutive letters — in the lines of the input file. It does not work properly, however. What does it actually do and what are the causes of the problems? Rewrite the procedure to work properly.

```
procedure genword()
   local line
   static letters
   initial letters := &lcase ++ &ucase
   while line := read() do
      scan line using
         while tab(upto(letters))
            do suspend tab(many(letters))
end
```

Acknowledgements

Although I author this Newsletter, several other persons are actively involved in the Icon project. Those presently active are Tim Budd (Ccon), Cary Coutant (Ccon, high-level string processing, language design and implementation), Dave Hanson (language design and implementation), and Steve Wampler (control structures, language design and implementation).

Request for Icon Documents

Please send the documents checked below to:

_____

_____

_____

_____

_____

- ☐ Corrections to Version 2.0 of Icon
- ☐ Corrections, Changes, and Known Bugs Related to Version 3.2 of Icon
- ☐ Corrigenda to the Icon Version 3 Reference Manual (TR 80-2)
- ☐ Expression Evaluation in Icon, TR 80-21
- ☐ Pattern Matching in Icon, TR 80-25
- ☐ "The Use of Character Sets and Character Mappings in Icon", reprinted from *Computer Journal*, May 1980
- ☐ Icon Observed Coding Laws and Standard Techniques (IconOCLAST)

Return this form to:

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

Portable Icon Distribution Request; Version 2.0

Contact Information:

name: _____

address: _____

_____

_____

_____

telephone: _____

cable/telex: _____

Computer Information:

manufacturer: _____

model: _____

memory capacity: _____

operating system: _____

comments: _____

_____

_____

_____

Magnetic Tape Information:

Our standard tape format distribution of the portable Icon system is 9-track, 1600 bpi (phase encoded), EBCDIC industry standard, unlabeled, fixed-blocked 80-character records with a blocking factor of 10 (last block filled out to 800 characters). Please indicate if you can accept this format.

☐ yes ☐ no

If you cannot accept this format, please indicate acceptable alternatives by checking the boxes below. Circle the checks that correspond to your preferred format. (You may fill out this section even if you can accept out standard format — we will try to accommodate your preferences, although doing so may cause delays.)

☐ 9-track	☐ 7-track	
☐ 1600 bpi	☐ 800 bpi	☐ 556 bpi
☐ EBCDIC	☐ ASCII	

blocking factor:

☐ 1	☐ 10	☐ other (specify) _____

comments:

☐ check here if you also want the UA Ratfor system

Return this form to:

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

Include either a magnetic tape or a check for $15.00 payable to the University of Arizona.

DEC-10 Icon Distribution Request; Version 2.0

Contact Information:

name: _____

address: _____

_____

_____

_____

telephone: _____

cable/telex: _____

Computer Information:

model: _____

memory capacity: _____

operating system: _____

comments: _____

_____

_____

_____

Magnetic Tape Information:

Icon for the DEC-10 is distributed as a BACKUP tape in interchange mode. Please specify your preferred tape recording format:

☐ 9-track ☐ 7-track
☐ 1600 bpi ☐ 800 bpi

comments:

Return this form to:

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

Include either a magnetic tape or a check for $15.00 payable to the University of Arizona.

CDC 6000/Cyber Icon Distribution Request; Version 2.0

Contact Information:

name: _____

address: _____

_____

_____

_____

telephone: _____

cable/telex: _____

Computer Information:

model: _____

memory capacity: _____

operating system: _____

character set: ☐ 63 ☐ 64

comments: _____

_____

_____

_____

Magnetic Tape Information:

Icon for CDC/Cyber systems is distributed as UPDATE PLs on an unlabeled SCOPE-format tape. Please specify your preferred tape recording characteristics:

☐ 9-track ☐ 7-track
☐ 1600 bpi ☐ 800 bpi ☐ 556 bpi

Return this form to:

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

Include either a magnetic tape or a check for $15.00 payable to the University of Arizona.

UNIX Icon Distribution Request; Version 3.2

Note: Version 3.2 of Icon is designed to run under Version 7 of UNIX on PDP-11 computers that have separate I and D spaces.

Contact Information:

name: _____

address: _____

_____

_____

_____

telephone: _____

cable/telex: _____

Magnetic Tape Information:

Please specify your preferred tape recording characteristics:

☐ 1600 bpi ☐ 800 bpi
☐ *tar* format ☐ *tp* format

Return this form to:

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

Include either a magnetic tape or a check for $15.00 payable to the University of Arizona.