



NewsLetter

Serving the Pascal, Modula-2, and Portable Programming Community

Vol. 4 No. 5 Sep - Dec 1990

IN THIS ISSUE

| | |
|---|----|
| Pascal Random-Number Generation and Testing | 1 |
| The Night Before Implementation of Life/70 | 7 |
| The Icon Language | 8 |
| An Overview of Version 8 of the Icon Programming Language | 9 |
| Preliminary Specification For A File Archiving Data Structure | 14 |
| A Modula-2 Bibliography | 21 |
| USUS 1991 Elections | 23 |
| Board Meeting Minutes (June 27, 1990) | 24 |
| Board Meeting Minutes (August 8, 1990) | 25 |
| Board Meeting Minutes (September 12, 1990) | 27 |
| Board Meeting Minutes (October 10, 1990) | 28 |
| Board Meeting Minutes (November 14, 1990) | 30 |
| Treasurer's Report | 32 |
| USUS 1991 Election Ballot | 33 |
| From The Editor | 35 |

Pascal Random-Number Generation and Testing

David T. Craig
736 Edgewater, Wichita, Kansas 67230
22 August 1990

Random-numbers play a very important role in today's computers. They appear in simulation programs to arcade games. Many computer languages and function libraries provide functions that return random values. But many of these random-number generators (RNG) are not very good. This paper describes a RNG that is supposed to be quite good. This generator was described in detail in BYTE magazine (March 1987, page 127+).

A simple Macintosh MPW Pascal program is provided that contains the BYTE RNG and a routine that tests the RNG using the Chi-Square statistical test (see Robert Sedgewick's wonderful book *Algorithms* for a detailed discussion of how to test a RNG). This program should easily be portable to other Pascal dialects.

The heart of the program is a function called `my_Random`. This function, as its name implies, returns a random integer in the range -32768 to +32767. To use it you must first initialize a record containing several seed values. These seeds start the random-number generation process. To generate a specific set of random-numbers initialize the seed values to known values. The other routine in the program is the `Test_RNG` procedure. This procedure tests the RNG using the Chi-Square test. Sedgewick states that this test should fail about 10% of the time.

When you run this program you should see results similar to those in figure 1.

A list of 100 random-numbers appears followed by 10 tests of 10,000 random-numbers. If the number sequence passes the Chi-Square test, then the numbers are statistically random.

The complete program including the routines `my_Random` and

Copyright 1990, USUS INC. All Rights Reserved.

The USUS NewsLetter is published ~6 times per year by USUS, the UCSD Pascal System User's Society, P.O. Box 1148 La Jolla, California 92038. The NewsLetter is a direct benefit of membership in USUS.

Tom Catrall Editor
William Smith Publisher

Test_RNG are in figure 2. Note that one change was made to the generator as published in BYTE. Instead of returning a real value in the range 0.0 to 1.0 my version returns an integer value. This change makes the generator faster since no floating point operations are needed.

I use this generator as a part of my programming bag of tricks. Note that the original authors of this RNG spent 3 months developing the generator.

That's all, folks ...

Figures 3 and 4 show two similar random number generators. They are from algorithms presented in the June 1988 issue of the Communications of the ACM. Figure 3 is for systems having 16 bit integers, and Figure 4 is for systems with 32 bit integers.

Figure 1

```
[ 1] -11876  31075 -28240  1827  -7314  2932  -11483  -9096
[ 9] -32668 -15103  27910  -3328 -27595  666  -15805  5520
[17]  27653  -954  -3997  -24345  32490  8653  18299  -22245
[25] -18133 -32258 -25647 -15150  20475 -30121  31560 -28408
[33] -32574  7313  11117  30479 -16649  -909  -14921  -2962
[41] -14991 -12705 -25744 -25355 -30740  31773  24876  20694
[49]  27721 -13539 -13371 -32638 -18938 -18239  2254  -18362
[57]  25578 -18604  18882 -22606  10289 -25462 -21118 -31008
[65]   9232  -8398  17661  29216 -21043  4687  -29783 -11655
[73] -23980 -5080  -6393 -12793  -7259  21908  11125 -32589
[81]  27915  -7411 -14237 -20603  -8381 -16311  -7459 -11699
[89] -19094 -22838  26388  -7016 -22203  5515  -7511  31798
[97] -17505 -29684 -29616 -22109

[ 1] Chi-Square value = 978      RNG PASSED TEST
[ 2] Chi-Square value = 982      RNG PASSED TEST
[ 3] Chi-Square value = 1087     RNG failed test <?>
[ 4] Chi-Square value = 937      RNG PASSED TEST
[ 5] Chi-Square value = 987      RNG PASSED TEST
[ 6] Chi-Square value = 994      RNG PASSED TEST
[ 7] Chi-Square value = 1054     RNG PASSED TEST
[ 8] Chi-Square value = 990      RNG PASSED TEST
[ 9] Chi-Square value = 1043     RNG PASSED TEST
[10] Chi-Square value = 1000     RNG PASSED TEST
```

Number of PASSES in 10 tries = 9
 NOTE: On the average this test should fail 1 out of 10 times.

Figure 2

```
PROGRAM Random_Number_Generator;

{$R+ } { [MPW PASCAL] enable range checking }
{$OV-} { [MPW PASCAL] disable overflow checking for routine my_Random() }

TYPE
  gt_RandomSeed = RECORD { seed(s) for the RNG }
    seed_x,seed_y,seed_z : INTEGER;
  END;
  gt_BigInteger = LONGINT; { > MAXINT (UCSD Pascal use INTEGER[12]) }

VAR
  gv_RandomSeed : gt_RandomSeed; { RNG seeds }
  gv_RNG_Index  : INTEGER; { indexer for the RNG }

{
  |||
  Routine .... my_Random
  Purpose .... Generate a random integer in the range -32768..32767
  Input ..... (none)
  Output ..... my_Random - random value
  Reference .. Byte Magazine, March 1987, p. 127+
  |||
}
```



```

BEGIN
  { clear the RNG frequency list }
  FOR f_index := 0 TO k_MaxFreqIndex DO rng_f_list[f_index] := 0;

  { compute the frequencies for a sample RNG run }
  FOR n_index := 1 TO k_NumCount DO
    BEGIN
      test_value := ABS( my_Random MOD (k_MaxFreqIndex + 1) );
      rng_f_list[test_value] := rng_f_list[test_value] + 1;
    END; { FOR n_index }

  { compute the chi-square stat for the generated RNG frequencies }
  t := 0;
  FOR f_index := 0 TO k_MaxFreqIndex DO
    BEGIN
      f_sqr := rng_f_list[f_index] * rng_f_list[f_index];
      t := t + f_sqr;
    END;

  chi_square := ((t * k_MaxFreqIndex) DIV k_NumCount) - k_NumCount;

  { tell the user the result of the chi-square test }
  WRITE(['',rng_test_loop:2,'] Chi-Square value = ',chi_square:6);
  IF (chi_square >= good_min_cs) AND (chi_square <= good_max_cs) THEN
    BEGIN
      WRITELN('          RNG PASSED TEST');
      good_count := good_count + 1;
    END
  ELSE
    WRITELN('          RNG failed test <?>');
  END; { FOR rng_test_loop }

  { tell the user the final result of all the RNG test runs }
  WRITELN;
  WRITELN('Number of PASSES in 10 tries = ',good_count:1);
  WRITELN;
  WRITELN('NOTE: On the average this test should fail 1 out of 10 times.');
```

```

END; { ----- Test_RNG ----- }

BEGIN { ----- Random_Number_Generator ----- }
  { initialize the random-number generator by setting the RNG seeds      }
  { to some unknown values (I use the Macintosh system clock via TickCount) }

  gv_RandomSeed.seed_x := TickCount MOD 30000;
  gv_RandomSeed.seed_y := TickCount MOD 1234;
  gv_RandomSeed.seed_z := TickCount MOD 5678;

  FOR gv_RNG_Index := 0 TO 99 DO { display a 100 random values }
    BEGIN
      IF gv_RNG_Index MOD 8 = 0 THEN
        BEGIN
          WRITELN;
          WRITE(['',(gv_RNG_Index+1):4,'] ');
          END;

          WRITE(my_Random:6, ' ');
        END;

      WRITELN; WRITELN;

      Test_RNG; { test the RNG }
    END. { ----- Random_Number_Generator ----- }

```

Figure 3

PROGRAM RandomTest;

VAR
s1, s2, s3 : INTEGER;
i, j : LONGINT;
bin : ARRAY[0..10] OF INTEGER;

FUNCTION Random16 : REAL;

{
Random number generator for systems having 16 bit integers.
The values s1, s2 and s3 need to be defined globally so that
their values persist between calls to Random16.

The generator is initialized by storing seed values into s1 and
s2.

This generator is from the article:

Efficient and Portable Combined Random Number Generators
by Pierre L'Ecuyer
June 1988 Communications of the ACM Page 742

The author of the article performed extensive statistical tests
(over 200 hours of VAX 11/780 cpu time) to verify the quality of
this generator.

VAR
z, k : INTEGER;
BEGIN
k := s1 DIV 206;
s1 := 157 * (s1 - k * 206) - k * 21;
IF s1 < 0 THEN
s1 := s1 + 32363;
k := s2 DIV 217;
s2 := 146 * (s2 - k * 217) - k * 45;
IF s2 < 0 THEN
s2 := s2 + 31727;
k := s3 DIV 222;

s3 := 142 * (s3 - k * 222) - k * 133;
IF s3 < 0 THEN
s3 := s3 + 31657;

z := s1 - s2;
IF z > 706 THEN
z := z - 32362;
z := z + s3;
IF z < 1 THEN
z := z + 32362;

Random16 := z * 3.0899E-5;
END; { Random16 }

BEGIN
Write('Enter s1 '); Readln(s1);
Write('Enter s2 '); Readln(s2);
Write('Enter s3 '); Readln(s3);

{ Print out a sample of random values }
FOR i := 1 TO 10 DO
WriteLn(Random16:16:12);

{ Do a simple minded test to verify that nothing has been lost in
transit.

The results should be that bin[0] .. bin[9] should contain
approx. 1000,
and bin[10] should contain 0.

FOR i := 0 TO 10 DO
bin[i] := 0;
FOR i := 1 TO 10000 DO
BEGIN
j := TRUNC(10.0 * Random16);
bin[j] := bin[j] + 1;
END;
FOR i := 0 TO 10 DO
WriteLn(i:4, ' ', bin[i]:5);
END.

Figure 4

```
PROGRAM RandomTest;
VAR
  s1, s2 : LONGINT;
  i, j   : LONGINT;
  bin    : ARRAY[0..10] OF INTEGER;
FUNCTION Random32 : REAL;
{
  Random number generator for systems having 32 bit LONGINTs.
  The values s1 and s2 need to be defined globally so that their
  values persist between calls to Random32.
  The generator is initialized by storing seed values into s1 and
  s2.
  This generator is from the article:
  Efficient and Portable Combined Random Number Generators
  by Pierre L'Ecuyer
  June 1988 Communications of the ACM Page 742
  The author of the article performed extensive statistical tests
  (over 200 hours of VAX 11/780 cpu time) to verify the quality
  of
  this generator.
}
VAR
  z, k : LONGINT;
BEGIN
  k := s1 DIV 53668;
  s1 := 40014 * (s1 - k * 53668) - k * 12211;
  IF s1 < 0 THEN
    s1 := s1 + 2147483563;
  k := s2 DIV 52774;
```

```

s2 := 40692 * (s2 - k * 52774) - k * 3791;
IF s2 < 0 THEN
  s2 := s2 + 2147483399;
z := s1 - s2;
IF z < 1 THEN
  z := z + 2147483562;
Random32 := z * 4.656613E-10;
END; { Random32 }
BEGIN
  Write('Enter s1 '); Readln(s1);
  Write('Enter s2 '); Readln(s2);
{ Write out a sample of random values }
  FOR i := 1 TO 10 DO
    Writeln(Random32:16:12);
  }
{ Do a simple minded test to verify that nothing has been lost in
  transit.
  The results should be that bin[0] .. bin[9] should contain
  approx. 1000,
  and bin[10] should contain 0.
}
  FOR i := 0 TO 10 DO
    bin[i] := 0;
  FOR i := 1 TO 10000 DO
    BEGIN
      j := TRUNC( 10.0 * Random32);
      bin[j] := bin[j] + 1;
    END;
  FOR i := 0 TO 10 DO
    Writeln(i:4, ' ', bin[i]:5);
  END.
END.
```

The Night Before Implementation of Life/70

Michael Robey

'twas the night before implementation, and all through the house,
not a program was working, not even a browse.

The programmers hung by their tubes in despair,
with hopes that a miracle soon would be there.

The users were nestled all snug in their beds,
while visions of inquiries danced in their heads.

When, out in the core, there arose such a clatter,
I sprang from my chair to see what was the matter.

And what, to my wondering eyes, should appear,
but a superprogrammer (with a six pack of beer).

His resume glowed with experience rare,
he churned out clean code with a bit-pusher's flair.

More rapid than eagles his programs, they came,
and he whistled, and shouted, and called them by name:

On backup! On update! On add! On delete!
On batch job! On closing! On function complete!

His eyes were glazed over, his belly quite lean,
from weekends and nights spent in front of a screen.

But a wink of his eye and a twist of his head,
soon gave me the know i had nothing to dread.

He spoke not a word, but went straight to his work,
turning specs into code, then turned with a jerk.

And, taking his finger from aside of his nose,
he pushed the key 'enter'. The system arose.

The updates updated; the deletes all deleted;
the inquiries inquired; and closings completed.

He tested each whistle and tested each bell,
with a nary an abend. All had gone well.

The system was ready, conversions concluded,
the user's last changes were even included.

Then the user exclaimed, with a snarl and a taunt,
"It's just what I asked for, but not what I want!"

The Icon Language

by Tom Cattrall

The following article on Icon came about after I happened upon the Icon bulletin board system when someone gave its number to me by mistake. Curiosity prompted me to look into Icon and I discovered an interesting string processing language that is available on many systems.

You can read the overview article to get a feel for what the language is like. If you wish to get a copy of Icon, you can download it for many systems from the Icon BBS, or order disks and/or tapes from the Icon project office at the University of Arizona.

The Icon project BBS number is: (602) 621-2283. It is also available by anonymous ftp: cs.arizona.edu (cd /icon)

Material available is listed below. All of it is available by mail or phone order. Much of it is available from the BBS (exceptions include Icon for the big IBM and DEC systems, and of course, the books and newsletters).

Icon Project
Department of Computer Science
Gould-Simpson Building
The University of Arizona
Tucson Arizona 85721
Phone: (602) 621-8448
Fax: (602) 621-4246

Icon Executables are available for the following systems. Prices are \$15 for all but the 2 MSDOS versions which are \$20 and \$25. These allow you to compile and run Icon programs on your system.

Amiga
Atari ST
MS-DOS
MS-DOS/386
Macintosh (MPW)
OS/2
UNIX PC
Xenix
Xenix/386

Source code is available for the following systems. The generic source code allows you to port Icon to a new environment.

Amiga
Atari ST
Generic for Porting (C source code I believe)
MS-DOS and OS/2
Macintosh (MPW)

The Icon Program Library is a collection of Icon programs that have been collected and made available in the following formats:

Amiga
Atari ST
MS-DOS and OS/2
Macintosh (MPW)
UNIX

Complete Systems for UNIX and large IBM and DEC machines is available on tape. These aren't available on the BBS system due to their size. (The UNIX system might be on the BBS)

CMS (IBM)
MVS (IBM)
UNIX
VMS (DEC)

Books and newsletters:

The Icon Programming Language \$34
The Implementation of Icon \$45
Icon Programming for Humanists \$30

The Icon Newsletter (back issues, 1 - 33) \$15
The Icon Newsletter (single issues) \$1
The Icon Analyst (more technical publication) \$25 / yr

An Overview of Version 8 of the Icon Programming Language*

Ralph E. Griswold

TR 90-6a

January 1, 1990

last revised February 13, 1990

Department of Computer Science

The University of Arizona

Tucson, Arizona 85721

1. Introduction

Icon is a high-level programming language with extensive facilities for processing strings and structures. Icon has several novel features, including expressions that may produce sequences of results, goal-directed evaluation that automatically searches for a successful result, and string scanning that allows operations on strings to be formulated at a high conceptual level.

Icon emphasizes high-level string processing and a design philosophy that allows ease of programming and short, concise programs. Storage allocation and garbage collection are automatic in Icon, and there are few restrictions on the sizes of objects. Strings, lists, and other structures are created during program execution and their size does not need to be known when a program is written. Values are converted to expected types automatically; for example, numeral strings read in as input can be used in numerical computations without explicit conversion. Icon has an expression-based syntax with reserved words; in appearance, Icon programs resemble those of Pascal and C.

Although Icon has extensive facilities for processing strings and structures, it also has a full repertoire of computational facilities. It is suitable for a wide variety of applications. Some examples are:

- text analysis, editing, and formatting
- document formatting
- artificial intelligence
- expert systems

*This work was supported by the National Science Foundation under Grant CCR-8713690.

- rapid prototyping
- symbolic mathematics
- text generation
- data laundry

There are public-domain implementations of Icon for the Amiga, the Atari ST, the Macintosh under MPW, MS-DOS, MVS, OS/2, many UNIX systems, VM/CMS, and VMS. There also is a commercial implementation of an enhanced version of Icon for the Macintosh [1].

The remainder of this report briefly describes the highlights of Icon. For a complete description, see [2,3].

2. Expression Evaluation

2.1 Conditional Expressions

In Icon there are conditional expressions that may succeed and produce a result, or may fail and not produce any result. An example is the comparison operation

$i > j$

which succeeds (and produces the value of j) provided that the value of i is greater than the value of j , but fails otherwise. Similarly,

$i > j > k$

succeeds if the value of j is between i and k .

The success or failure of conditional operations is used instead of Boolean values to drive control structures in Icon. An

example is

```
if i > j then k := i else k := j
```

which assigns the value of *i* to *k* if the value of *i* is greater than the value of *j*, but assigns the value of *j* to *k* otherwise.

The usefulness of the concepts of success and failure is illustrated by `find(s1,s2)`, which fails if *s1* does not occur as a substring of *s2*. Thus

```
if i := find("or",line) then write(i)
```

writes the position at which "or" occurs in line, if it occurs, but does not write a value if it does not occur.

Many expressions in Icon are conditional. An example is `read()`, which produces the next line from the input file, but fails when the end of the file is reached. The following expression is typical of programming in Icon and illustrates the integration of conditional expressions and conventional control structures:

```
while line := read() do  
write(line)
```

This expression copies the input file to the output file.

If an argument of a function fails, the function is not called, and the function call fails as well. This "inheritance" of failure allows the concise formulation of many programming tasks. Omitting the optional `do` clause in `while-do`, the previous expression can be rewritten as

```
while write(read())
```

2.2 Generators

In some situations, an expression may be capable of producing more than one result. Consider

```
sentence := "Store it in the neighboring harbor"  
find("or",sentence)
```

Here "or" occurs in sentence at positions 3, 23, and 33. Most programming languages treat this situation by selecting one of the positions, such as the first, as the result of the expression. In Icon, such an expression is a generator and is capable of producing all three positions.

The results that a generator produces depend on context. In a situation where only one result is needed, the first is produced, as in

```
i := find("or",sentence)
```

which assigns the value 3 to *i*.

If the result produced by a generator does not lead to the success of an enclosing expression, however, the generator is resumed to produce another value. An example is

```
if (i := find("or",sentence)) > 5 then write(i)
```

Here the first result produced by the generator, 3, is assigned to *i*, but this value is not greater than 5 and the comparison operation fails. At this point, the generator is resumed and produces the second position, 23, which is greater than 5. The comparison operation then succeeds and the value 23 is written. Because of the inheritance of failure and the fact that comparison operations return the value of their right argument, this expression can be written in the following more compact form:

```
write(5 < find("or",sentence))
```

Goal-directed evaluation is inherent in the expression evaluation mechanism of Icon and can be used in arbitrarily complicated situations. For example,

```
find("or",sentence1) = find("and",sentence2)
```

succeeds if "or" occurs in sentence1 at the same position as and occurs in sentence2.

A generator can be resumed repeatedly to produce all its results by using the `every-do` control structure. An example is

```
every i := find("or",sentence)  
do write(i)
```

which writes all the positions at which "or" occurs in sentence. For the example above, these are 3, 23, and 33.

Generation is inherited like failure, and this expression can be written more concisely by omitting the optional `do` clause:

```
every write(find("or",sentence))
```

There are several built-in generators in Icon. One of the most frequently used of these is

```
i to j
```

which generates the integers from *i* to *j*. This generator can be combined with `every-do` to formulate the traditional `for-style` control structure:

```
every k := i to j do  
f(k)
```

Note that this expression can be written more compactly as

```
every f(i to j)
```

There are a number of other control structures related to generation. One is alternation,

expr1 | expr2

which generates the results of expr1 followed by the results of expr2. Thus

```
every write(find("or",sentence1) | find("or",sentence2))
```

writes the positions of "or" in sentence1 followed by the positions of "or" in sentence2. Again, this sentence can be written more compactly by using alternation in the second argument of find():

```
every write(find("or",sentence1 | sentence2))
```

Another use of alternation is illustrated by

```
(i | j | k) = (0 | 1)
```

which succeeds if any of i, j, or k has the value 0 or 1.

Procedures can be used to add generators to Icon's built-in repertoire. For example,

```
procedure findodd(s1,s2)
  every i := find(s1,s2) do
    if i % 2 = 1 then suspend i
end
```

is a procedure that generates the odd-valued positions at which s1 occurs in s2. The suspend control structure returns a value from the procedure, but leaves it in suspension so that it can be resumed for another value. When the loop terminates, control flows off the end of the procedure without producing another value.

3. String Scanning

For complicated operations, the bookkeeping involved in keeping track of positions in strings becomes burdensome and error prone. Icon has a string scanning facility that manages positions automatically. Attention is focused on a current position in a string as it is examined by a sequence of operations.

The string scanning operation has the form

```
s ? expr
```

where s is the subject string to be examined and expr is an expression that performs the examination. A position in the subject, which starts at 1, is the focus of examination.

Matching functions change this position. One matching function, move(i), moves the position by i and produces the substring of the subject between the previous and new positions. If the position cannot be moved by the specified amount (because the subject is not long enough), move(i) fails. A simple example is

```
line ? while write(move(2))
```

which writes successive two-character substrings of line, stopping when there are no more characters. Another matching function is tab(i), which sets the position in the subject to i and also returns the substring of the subject between the previous and new positions. For example,

```
line ? if tab(10) then write(tab(0))
```

first sets the position in the subject to 10 and then to the end of the subject, writing line[10:0]. Note that no value is written if the subject is not long enough.

String analysis functions such as find() can be used in string scanning. In this context, the string that they operate on is not specified and is taken to be the subject. For example,

```
line ? while write(tab(find("or")))
do move(2)
```

writes all the substrings of line prior to occurrences of "or". Note that find() produces a position, which is then used by tab to change the position and produce the desired substring. The move(2) skips the "or" that is found.

Another example of the use of string analysis functions in scanning is

```
line ? while tab(upto(&letters)) do
write(tab(many(&letters)))
```

which writes all the words in line.

As illustrated in the examples above, any expression may occur in the scanning expression.

4. Structures

Icon supports several kinds of structures with different organizations and access methods. Lists are linear structures that can be accessed both by position and by stack and queue functions. Sets are collections of arbitrary values with no implied ordering. Tables provide an associative lookup mechanism.

4.1 Lists

While strings are sequences of characters, lists in Icon are sequences of values of arbitrary types. Lists are created by enclosing the lists of values in brackets. An example is

```
car1 := ["buick", "skylark", 1978, 2450]
```

in which the list car1 has four values, two of which are strings and two of which are integers. Note that the values in a list

need not all be of the same type. In fact, any kind of value can occur in a list - even another list, as in

```
inventory := [car1,car2,car3,car4]
```

Lists also can be created by

```
L := list(i,x)
```

which creates a list of *i* values, each of which has the value *x*.

The values in a list can be referenced by position much like the characters in a string. Thus

```
car1[4] := 2400
```

changes the last value in *car1* to 2400. A reference that is out of the range of the list fails. For example,

```
write(car1[5])
```

fails.

The values in a list *L* are generated by *!L*. Thus

```
every write(!L)
```

writes all the values in *L*.

Lists can be manipulated like stacks and queues. The function *push(L,x)* adds the value of *x* to the left end of the list *L*, automatically increasing the size of *L* by one. Similarly, *pop(L)* removes the leftmost value from *L*, automatically decreasing the size of *L* by one, and produces the removed value.

4.2 Sets

A set is a collection of values. An empty set is created by *set()*. Alternatively, *set(L)* produces a set with the values in the list *L*. For example,

```
S := set([1,"abc",[ ]])
```

assigns to *S* a set that contains the integer 1, the string "abc", and an empty list.

The set operations of union, intersection, and difference are provided. The function *member(S,x)* succeeds if *x* is a member of the set *S* but fails otherwise. The function *insert(S,x)* adds *x* to the set *S*, while *delete(S,x)* removes *x* from *S*. A value only can occur once in a set, so *insert(S,x)* has no effect if *x* is already in *S*. *!S* generates the members of *S*.

A simple example of the use of sets is given by the following segment of code, which lists all the different words that appear in the input file:

```
words := set()
while line := read() do
```

```
line ? while tab(upto(&letters)) do
    insert(words,tab(many(&letters)))
every write(!words)
```

4.3 Tables

Tables are sets of pairs, each of which consists of a key and a corresponding value. The key and its corresponding value may be of any type, and the value for any key can be looked up automatically. Thus, tables provide a form of associative access in contrast with the positional access to values in lists.

A table is created by an expression such as

```
symbols := table(0)
```

which assigns to *symbols* a table with the default value 0. The default value is used for keys that are not assigned another value. Subsequently, *symbols* can be referenced by any key, such as

```
symbols["there"] := 1
```

which assigns the value 1 to the key "there" in *symbols*.

Tables grow automatically as new keys are added. For example, the following program segment produces a table containing a count of the words that appear in the input file:

```
words := table(0)
while line := read() do
    line ? while tab(upto(&letters)) do
        words[tab(many(&letters))] += 1
```

Here the default value for each word is 0, as given in *table(0)*, and *+=* is an augmented assignment operation that increments the values by one. There are augmented assignment operations for all binary operators.

A list can be obtained from a table *T* by the function *sort(T,i)*. The form of the list depends on the value of *i*. For example, if *i* is 3, the list contains alternate keys and their corresponding values in *T*. For example,

```
wordlist := sort(words,3)
while write(pop(wordlist)," : ",pop(wordlist))
```

writes the words and their counts from *words*.

5. An Example

The following program, which produces a concordance of the words from an input file, illustrates typical Icon programming techniques. Although not all the features in this program are described in previous sections, the general idea should be clear.

global uses, lineno, width

```
procedure main(args)
  width := 15          # width of word field
  uses := table()
  lineno := 0
  every tabulate(words()) # tabulate all the
                          # citations
  output()            # print the citations
end
```

```
# Add line number to citations for word
#
procedure tabulate(word)
  /uses[word] := table()
  uses[word][lineno] := 1
  return
end

# Generate words
#
procedure words()
  while line := read() do {
    lineno += 1
    write(right(lineno,6)," ",line)
    map(line) ? while tab(upto(&letters)) do {
      s := tab(many(&letters))
      if *s < 3 then next # skip short words
      suspend s
    }
  }
end

# Print the results
#
procedure output()
  write()          # blank line
  uses := sort(uses,3) # sort citations
  while word := get(uses) do {
    line := ""
    numbers := sort(get(uses),3)
    while line ||:= get(numbers) || ", " do
      get(numbers) # skip marking value
      write(left(word,width),line[1:-2])
    }
  }
end
```

The program reads a line, writes it out with an identifying line number, and processes every word in the line. Words less than three characters long are considered to be "noise" and are discarded. A table, uses, is keyed by the words. Every key has a corresponding set of line numbers. The first time a word is encountered, a new set is created for it. The line number is inserted in any event. Since a value can be in a set only once, duplicate line numbers are suppressed automatically.

After all the input has been read, the table of words is sorted by key. Each corresponding set of line numbers is sorted and the line numbers are appended to the line to be written.

For example, if the input file is

```
On the Future!-how it tells
Of the rapture that impells
To the swinging and the ringing
Of the bells, bells, bells-
Of the bells, bells, bells, bells,
  Bells, bells, bells-
To the rhyming and the chiming of the bells!
```

the output is

```
1   On the Future!-how it tells
2   Of the rapture that impells
3   To the swinging and the ringing
4   Of the bells, bells, bells-
5   Of the bells, bells, bells, bells,
6     Bells, bells, bells-
7   To the rhyming and the chiming of the bells!
```

```
and   3, 7
bells 4, 5, 6, 7
chiming 7
future 1
how 1
impells 2
rapture 2
rhyming 7
ringing 3
swinging 3
tells 1
that 2
the 1, 2, 3, 4, 5, 7
```

Acknowledgement

Icon was designed by the the author in collaboration with Dave Hanson, Tim Korb, Cary Coutant, and Steve Wampler. Many other persons have contributed to its development. The current implementation is based on the work of Cary Coutant and Steve Wampler with recent contributions by Bill Mitchell, Janalee O'Bagy, Gregg Townsend, and Ken Walker.

References

1. R. E. Griswold and M. T. Griswold, The ProIcon Programming Language for the Apple Macintosh Computers, The Bright Forest Company, Tucson, AZ, 1989.
2. R. E. Griswold and M. T. Griswold, The Icon Programming Language, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.
3. R. E. Griswold, Version 8 of Icon, The Univ. of Arizona Tech. Rep. 90-1, 1990.

PRELIMINARY SPECIFICATION FOR A FILE ARCHIVING DATA STRUCTURE

Prepared by David T. Craig
736 Edgewater, Wichita, Kansas 67230
24 March 1990

This specification was written in response to the group project comments put forth by Tom Catrall, editor of the USUS NewsLetter. This information is provided for public comment by the members of USUS. In the author's opinion, file data structures should be designed first, and only then should data access routines be written to access the file data.

See the section titled Modifications at the end of this document for a list of the latest modifications to this specification.

Ideas for this file structure were liberally borrowed from Microsoft's Tag Image File Format (TIFF), the UNIX file system, the Apple Lisa file system, UCSD Pascal object files, and an early archive file format that I developed several years ago but never finished.

This specification has the following goals:

- Package different types of files together into a single file
- Compress the file data
- Include the original file attributes
- Provide reliable and self-describing data structures
- Avoid as much as possible fixed locations for any data structures
- Be independent of any specific file system
- Be independent of any file byte ordering
- Be independent of any programming language or data types
- Provide fast access to any archived file data

0.0 Specification Notes

All file structure diagrams include the byte size of each component that is listed at the right side of each component description and is surrounded by [and] (eg: [4] means 4 bytes). Structures which have a variable size have the designation [V].

All files are viewed as a linear sequence of bytes with the first byte in the file called byte 0, the second file byte is called byte 1, etc.

All file offsets are relative to the start of the file, are positive (the sign bit is ignored), and are four bytes in length. This length gives the file a maximum size of 2,147,483,648 bytes. The sign bit is ignored since many high-level languages don't support unsigned integers. Offsets do not have to be word based. They may point to

any byte within the archive file.

The byte order of the file may either be Most Significant Byte first order or Least Significant Byte first order. Archive files specify which order is used. Note that the data files that are archived do not have any byte ordering associated with them. Only the archive file control values have an associated order.

Values prefaced by \$ are hexadecimal values (eg: \$10 is 16 decimal). Hexadecimal values used in this specification are written with the most significant byte first (eg: 256 decimal is written \$0100).

All strings consist of a sequence of character bytes prefaced by a single string length byte. The size of a string field normally is variable. For example, the string "David" would occupy only 6 bytes (1 length byte + 5 characters). Strings with a fixed field size are explicitly documented as having a fixed field size. Fixed size strings should have unused bytes set to zero. For example, if the fixed string field is 8 characters in size and the string is "DTC", then the first byte should contain the string length (3), the next 3 bytes should contain the characters "DTC", and the remaining 4 bytes should contain zero values. This is done since hex dumps of the file will look a lot better and be more easily understood if fixed strings are setup in this manner.

This specification occasionally uses phrase fields. Phrases are strings whose length is contained in 2 bytes and can contain from 0 to 32767 bytes. Phrases are used when strings are insufficient.

This specification does not currently use data types or language peculiarities that may cause problems. For example, no floating point values are used since no standard format exists. The same applies to data packing. Some languages support it, some don't, and others that do use different packing algorithms.

All areas which are reserved for future use must be cleared to zero.

Reference is made within this specification to the Archive Program. This is the actual program that manipulates archive files. Users will run this program since it provides a user interface. Another specification describes the functions of the Archive Program. Reference may also be made to the Archive File Manager. This is a low-level program that does not contain a user interface. The

Archive Program should call the Archive File Manager since this manager provides all the I/O control for archive files. Another specification describes the functions of the Archive File Manager.

1.0 General File Structure

Archive files contain three general areas:

- Header Area
- File Directory Area
- File Data Area

The Header Area contains basic information about the archive file. This area **MUST** exist at the beginning of the file. The other areas are pointed to by the Header and may exist at any place in the file. The File Directory Area may even occur after the File Data Area.

The Header Area and the File Directory Area are both duplicated in the archive file. The duplicate areas are called the Duplicate Header Area and the Duplicate Directory Area. The duplicates exist in case the primary areas become corrupt. A Scavenger program will locate the duplicate areas and rebuild the archive file from this duplicate information.

The Header Area consists of a single record. The File Directory Area consists of multiple records. The File Data Area consists also of multiple records. Given that the Directory and Data areas may occur anywhere in the archive file and these two areas consist of multiple records, the Directory and Data area records may be intermixed within the archive file. Therefore, the above diagram represents **ONLY** a figurative arrangement of the information within an archive file; the physical information arrangement may differ dramatically from the diagram.

Gaps may exist within the file. For example, an implementation may base all file records on 512 byte boundaries to maximize I/O speed for disk drives that are based upon 512 byte sectors. Any gaps that do exist should be filled with zeros for consistency sake, even though the actual values within the gaps are immaterial.

1.1 General File Structure: (Primary) Header Area

The Header is the only component of the archive file structure that must reside in a specific location within the file. All other components may reside anywhere since they are pointed to by offsets within other archive file components.

The Header **MUST** exist at the beginning of the file. If the Header data does not exist there, then the file is corrupt.

The Header Area has the structure:

| | |
|------------------------------------|------|
| Unique ID (\$31415926) | [4] |
| File Byte Order ("LL" or "MM") | [2] |
| File Version | [2] |
| Header Flags | [2] |
| File Directory Area Offset | [4] |
| Archive File Password | [8] |
| Archive File Comment Record Offset | [4] |
| Duplicate Header Area Offset | [4] |
| Omega Record Offset | [4] |
| Reserved for Future Use | [16] |
| Header Record CRC | [2] |
| Total: | 52 |

Unique ID:

A (hopefully) unique value that is used to verify that this file is really an archive file. All programs that access archive files must check that these bytes exist as the first four bytes in the file. (contains the digits of Pi)

File Byte Order:

This field describes the byte order of all multiple byte fields within the file. If the field contains the characters "LL" (\$4C4C), then the least significant byte (LSB) occurs first. If the field contains the characters "MM" (\$4D4D), then the most significant byte (MSB) occurs first. All programs that access these archive files should support both LSB and MSB data formats.

File Version:

The version number of the file. The current version number is 0 (\$0000). Future versions will increment this value by one (eg: version 1 files contain \$0001). Note: I avoided the TIFF version number of 42 even though Arthur Dent and Ford Prefect will be very disappointed.

Header Flags:

Set of 16 bit flags with the following meanings:

- Bit 0 : Archive file DIRTY flag
- Bits 1-15 : Reserved for future use (clear to all zeros)

Bit 0 is very special for the integrity of the archive file. When the Archive Program opens an archive file, the program should set this bit since the file is open and is considered "dirty". When the Archive Program closes the archive file, the program should clear this bit. If the Archive Program, during the processing of an archive file, crashes, then this bit will be set. When the Archive Program starts, it should check this bit. If the bit is set, then the archive file is dirty and should be scavenged. If the bit is cleared, then the archive file was properly closed by the Archive Program. (this idea is based upon Microsoft's HPFS which uses a similar bit for file system integrity)

File Directory Area Offset:

Offset to the File Directory Area.

Archive File Password:

Encrypted password for the file. The encryption algorithm should be simple. I recommend a simple Exclusive-OR algorithm using a "secret" encryption key. If this field contains all zeros, then no password exists and any program can access the file data.

Archive File Comment Record Offset:

Offset to the comment record for use by the Archive Program user. See section 1.1.2 for the record details.

Duplicate Header Area Offset:

Offset to the Duplicate Header Area. This area has the exact same format as the Primary Header Area. Whenever the Primary Header Area is altered, the Duplicate should be too. The Duplicate Header Area should reside far away from the Primary Header Area for data integrity. See section 1.1.1 for the details behind the Duplicate Header Area.

Omega Record Offset:

Offset to the special Omega Record. This record marks the logical end of the archive file. See section 1.1.3 for the details.

Reserved for Future Use:

This field exists since I'm not always correct when it comes to file data structures.

Header Record CRC:

CRC of all the Header data excluding this CRC field. The Archive Program should verify that the archive file has valid CRCs before performing data I/O.

1.1.1 Duplicate Header Area

The Duplicate Header Area is an almost exact duplicate copy of the Primary Header Area. This Area has its own unique ID (\$11447298) (contains ln Pi) that differs from the Primary Header Unique ID. The Duplicate Header Area Offset points to the Primary Header Area. Even though the Primary Header Area starts at offset 0, this location may change in the future and both Header Areas should reference each other.

This area exists in case the Primary Header Area becomes corrupt. A Scavenger program will then search the archive file looking for the Duplicate Header Area and when found will rebuild the Primary Header Area.

1.1.2 Archive File Comment Record

The Archive File Comment Record contains the archive file comment phrase that the user of the Archive Program may have entered. The record's format follows:

| | |
|--------------------------|------|
| Unique ID (\$98696044) | [4] |
| Record Length | [4] |
| Comment Length (0-32767) | [2] |
| Comment Characters | [V] |
| Record CRC | [2] |
| Total: | V |

Unique ID:

A unique ID that identifies this record as the Archive File Comment. (contains Pi squared)

Record Length:

Length in bytes of all the data stored within the record. This field is used for reliability reasons.

Comment Length:

Byte length of the comment phrase. If the length is zero, then the comment phrase is empty and the Comment Characters' field does not exist.

Comment Characters:

Comment phrase characters. This field may be up to 32K in length and does not contain a length byte as string fields do. This field was made large so that large comments could be stored with an archive file.

Record CRC:

CRC of all the record data excluding the CRC field itself.

1.1.3 Omega Record

This record marks the logical end of the archive file. It must exist at the very end of the archive file. The Scavenger will use this record when it attempts to locate the logical end of the archive file.

| | |
|------------------------|------|
| Unique ID (\$25806975) | [4] |
| Record Length | [4] |
| Record CRC | [2] |
| Total: | 10 |

Unique ID:

Unique ID that marks this record as the Omega Record. (contains the square root of 666)

Record Length:

Length in bytes of all the data in this record. For the Omega Record, this field contains the value 10.

Record CRC:

CRC of all the record data excluding the CRC field itself.

1.2 General File Structure: File Directory Area

The File Directory Area is pointed to by the File Header. This structure contains all the references to the archived files. The File Directory Area may contain multiple Directory Entries, one for each archived file.

The File Directory Area has the structure:

| | |
|---------------------------------|------|
| Unique ID (\$27182818) | [4] |
| Record Length | [4] |
| Number of Archived Files (NAF) | [2] |
| File Directory Entry # 1 | [12] |
| File Directory Entry # 2 | [12] |
| ... | |
| File Directory Entry # NAF | [12] |
| Duplicate Directory Area Offset | [4] |
| Reserved for Future Use | [16] |
| File Directory Area CRC | [2] |
| Total: | V |

Unique ID:

A unique value that is used to verify that this data is really the File Directory Area. (contains e)

Record Length:

Length in bytes of all the data stored within the record. This field is used for reliability reasons.

Number of Archived Files:

The number of files within the archive file. If this value is zero, then no files have been archived.

File Directory Entry:

Directory entry for a single file. See section 1.2.1 for the details.

Duplicate Directory Area Offset:

Offset to the Duplicate Directory Area. The Duplicate Area contains an almost exact copy of the Primary Directory Area. See section 1.2.3 for the details.

Reserved for Future Use:

This field is reserved for future use and should be cleared.

File Directory Area CRC:

A CRC of all the data in the File Directory Area excluding the CRC field itself.

1.2.1 File Directory Entry structure

This data structure is based upon the UNIX file system. As such, it basically contains pointers to the actual directory entry data. The File Directory Area may contain multiple Directory Entries, one for each archived file.

| | |
|----------------------------|------|
| File Name Record Offset | [4] |
| File Comment Record Offset | [4] |
| F-Node Offset | [4] |
| Total: | 12 |

File Name Record Offset:

Offset to the file name record. See section 1.2.2 for the details.

File Comment Record Offset:

Offset to the file comment record. See section 1.2.3 for the details.

F-Node Offset:

Offset to the actual file directory data. The phrase "F-Node" is borrowed from UNIX's i-nodes (I didn't want to steal UNIX's nomenclature so I used my own -- sorry AT&T). See section 1.2.4 for the details behind F-Nodes.

1.2.2 File Name Record

The File Name Record contains the file name phrase that the user of the Archive Program may have entered. The record's format follows:

| | |
|------------------------|------|
| Unique ID (\$31006276) | [4] |
| Record Length | [4] |
| F-Node Offset | [4] |
| File Name Length | [2] |
| File Name Characters | [V] |
| Record CRC | [2] |
| Total: | V |

Unique ID:

A unique ID that identifies this record as a File Name Record. (contains Pi cubed)

Record Length:

Length in bytes of all the data stored within the record. This field is used for reliability reasons.

F-Node Offset:

Offset of the F-Node structure for this record. This field is used for data integrity and recovery.

File Name Length:

Byte length of the name phrase. If the length is zero, then the name phrase is empty and the File Name Characters' field does not exist.

File Name Characters:

Comment phrase characters. This field may be up to 32K in length and does not contain a length byte as string fields do. This field was made large so that large comments could be stored with an archive file.

Record CRC:

CRC of all the record data excluding the CRC field itself.

1.2.3 File Comment Record

The File Comment Record contains the file comment phrase that the user of the Archive Program may have entered. The record's format follows:

| | |
|-------------------------|------|
| Unique ID (\$17724538) | [4] |
| Record Length | [4] |
| F-Node Offset | [4] |
| File Comment Length | [2] |
| File Comment Characters | [V] |
| Record CRC | [2] |
| Total: | V |

Unique ID:

A unique ID that identifies this record as a File Comment Record. (contains square root of Pi)

Record Length:

Length in bytes of all the data stored within the record. This field is used for reliability reasons.

F-Node Offset:

Offset of the F-Node structure for this record. This field is used for data integrity and recovery.

File Comment Length:

Byte length of the comment phrase. If the length is zero, then the comment phrase is empty and the Comment Characters' field does not exist.

File Comment Characters:

Comment phrase characters. This field may be up to 32K in length and does not contain a length byte as string fields do. This field was made large so that large comments could be stored with an archived file.

Record CRC:

CRC of all the record data excluding the CRC field itself.

1.2.4 F-Node structure

An F-Node (File-Node) contains the actual file directory data. Every archived file has its own unique F-Node. F-Node data is not incorporated directly into the File Directory Area for space reasons, I/O speed reasons, and reliability reasons.

| | |
|-----------------------------|------|
| Unique ID (\$23025851) | [4] |
| Record Length | [4] |
| File Directory Entry Offset | [4] |
| File Flags | [2] |
| File Data Record Offset | [4] |
| Compression Type | [2] |
| Compression File Handler | [64] |
| File Type | [2] |
| Private File Data Handler | [64] |
| Private File Data Length | [2] |
| Private File Data | [V] |
| Reserved for Future Use | [4] |
| Record CRC | [2] |
| Total: | V |

Unique ID:

A unique ID for every F-Node. (contains ln 10)

Record Length:

Length in bytes of the F-Node record.

File Directory Entry Offset:

Offset of the File Directory Entry for the F-Node. This field is used for data integrity and recovery.

File Flags

Bit flags provided for protection and access control. The bits and their meaning follow:

- Bit 0 - Read allowed when set to 1
- Bit 1 - Write allowed when set to 1

Bits 2-15 are reserved for future use and should be set to 0. Note that bit 0 is the least significant bit and bit 15 is the most significant bit (follows Motorola nomenclature).

ture).

File Data Record Offset:

Offset of the File Data Record.

Compression Type:

Type of compression that was performed on the File Data. Contains the following values:

- \$0000 - No compression
- \$0001 - PackBits compression (used by Macintosh Mac Paint)
- \$0002 - Huffman compression
- \$0003 - LZW compression
- \$0004 - Pascal Token compression
- \$0005 - Modula-2 Token compression
- \$0006 - Modula-3 Token compression
- \$0007 - ADA Token compression
- \$0008 - C Token compression
- \$0009 - BASIC Token compression

The PackBits compression scheme is used by MacPaint and has become a standard in the Macintosh community. This scheme is fully documented by Apple Computer and is simple to implement. Note that PackBits is best used on image files which contain repeated runs of bytes. Other compression schemes exist for more random data. (this is an issue for the Archive Program)

The Huffman and LZW schemes are fairly popular, but are not so simple to implement.

The Token compression schemes are proposed by the author of this specification since they are simple to implement and the author already has a Pascal program that tokenizes a Pascal source program.

Compression File Handler:

A string field containing the name of an executable program file that will be dynamically called when the File Data is to be compressed or decompressed. The executable file will be passed two file names; the original uncompressed file and the name of the file to hold the compressed file data.

This field and its physical working have not been fully defined. As such, you should consider this field as reserved for future use and should be filled with zeros.

If this field is used, then field Compression Type should become obsolete.

File Type:

The file type that corresponds to the computer file system that created the file originally. Contains the following values:

- \$0000 - Unknown
- \$0001 - Macintosh MFS
- \$0002 - Macintosh HFS
- \$0003 - IBM PC MS-DOS
- \$0004 - IBM PC HPFS
- \$0005 - UCSD P-System

Use the Unknown type (\$0000) if the file type is unimportant. If the file type is Unknown, then no Private File Data exists and the Private File Data Length will be zero. Only the Archive Program has knowledge of these types.

This field determines the kind and format of the data within the Private File Data field. For example, if the File Type is Macintosh MFS, then the Private File Data field could contain the standard Macintosh file TYPE and CREATOR identifiers.

Private File Data Handler:

A string field containing the name of an executable program file that will be dynamically called when the Private File Data field is to be setup or interpreted.

This field and its physical working have not been fully defined. As such, you should consider this field as reserved for future use and should be filled with zeros.

If this field is used, then field File Type should become obsolete.

Private File Data Length:

Length in bytes of the Private File Data.

Private File Data:

Data specific to the file type. For example, a Macintosh MFS file has a unique directory structure that could be stored here. Only the Archive Program has knowledge about what data should be stored here.

Reserved for Future Use:

This field is reserved for future use and should be cleared.

Record CRC:

CRC of all the fields in the F-Node record.

1.2.3 Duplicate Directory Area

The Duplicate Directory Area is an almost exact duplicate copy of the Primary Directory Area. This Area has its own unique ID (\$49714987) (contains log Pi) that differs from the Primary Directory Unique ID. The Duplicate Directory Area Offset points to the Primary Directory Area.

This area exists in case the Primary Directory Area

becomes corrupt. A Scavenger program will then search the archive file looking for the Duplicate Directory Area and when found will rebuild the Primary Directory Area.

This field is reserved for future use and should be cleared.

1.3 General File Structure: File Data Area

The File Data Area contains the actual file data. This area may contain multiple file data each with the following record structure:

| | |
|-----------------------------|------|
| Unique ID (\$43429448) | [4] |
| File Directory Entry Index | [2] |
| File Directory Entry Offset | [4] |
| F-Node Offset | [4] |
| File Data Length | [4] |
| File Data | [V] |
| File Data CRC | [2] |
| Reserved for Future Use | [4] |
| Record CRC | [2] |
| Total: | V |

Unique ID:

A-unique ID used for data integrity. (contains log e)

File Directory Entry Index:

Index of the File Directory Entry for this data. This field is used for data integrity and recovery. (relative to 1)

File Directory Entry Offset:

Offset of the File Directory Entry for this record. This field is used for data integrity and recovery.

F-Node Offset:

Offset of the F-Node structure for this data. This field is used for data integrity and recovery.

File Data Length:

Length in bytes of the File Data.

File Data:

Raw file data. May exist in a compressed format.

File Data CRC:

CRC value of only the File Data.

Reserved for Future Use:

Record CRC:

CRC of all the fields in the record excluding the CRC field itself.

2.0 Archive File Data Recovery

One goal of this specification is the provision for reliable and self-describing data structures. Reliability is achieved by storing redundant crucial data through out the archive file and providing links between these crucial data areas. Every data object in the archive file is self-identifying. This feature forms the foundation upon which the Scavenger program may rebuild corrupt archive files.

The Scavenger may be a stand-alone program or it may be part of the Archive File Manager. At this time not enough is known about the higher software levels of this project to answer these types of questions.

3.0 Summary

This specification may seem verbose to some programmers. But the author believes quite strongly that the verbosity is warranted if the archive file data structure is to be both flexible and recoverable from data corruption.

Any archive program should provide data recovery in case the archive file becomes corrupt. In a similar manner to the Apple Lisa file system, any detection of corruption should initiate a Scavenger process. The Scavenger attempts to rebuild the archive file and if this fails, then as much file data within the archive file as possible should be extracted and stored in separate files for future archiving.

Any positive or negative comments are welcome (the positive comments are naturally more welcome than the negative).

4.0 Modifications

24 Mar 1990: Rearranged many of the records and cleaned up the specification. Added the duplicate Header and Directory. Added more data links for recovery purposes.

A Modula-2 Bibliography

compiled by Dick Ezzard

Rechenberg, P. and H. Moessenboeck
A Compiler Generator for Microcomputers
Prentice Hall / Hanser Verlag, 1989

Pinson, Lewis et al
A First Course in Computer Science with Modula-2.
Wiley, 1987

Christian, Kaare
A guide to Modula-2
Springer-Verlag, c1986.

McCracken, Daniel D. & W. Salmon
A second course in computer science with Modula-2
Wiley, c1987.

Harrison, Rachael.
Abstract Data Types in Modula-2
John Wiley, 1989

Schildt, Herbert.
Advanced Modula-2
Osborne McGraw-Hill, c1987.

Ward, Terry A.
Advanced programming techniques in Modula-2
Scott, Foresman, c1987.

Wirth, Niklaus
Algorithms and data structures, 1986 edition
Prentice-Hall, 1986
("...major change... Pascal has been replaced by Modula-2.")

Adams, J. Mack et al
An Introduction to Computer Science with Modula-2.
Heath, 1988

Terry, Patrick D.
An introduction to programming with Modula-2
Addison-Wesley Pub. Co., c1987.

Tucker, Allen B., Jr.
Computer science : a second course using Modula-2
New York : McGraw-Hill, c1988.

Schipper, Andre; translated by Jack Howlett.
Concurrent programming : illustrated with examples in
Pascal, Modula-2, and Ada
New York : Halsted Press, c1989.

Riley, David D., Data Abstraction and Structures : an
introduction to computer science II
Boston : Boyd & Fraser Pub. Co., c1987.

Nyhoff, Larry and Sanford Leestma
Data Structures & Advanced Programming in Modula-2
Macmillan, 1990

Sincovec, Richard F. & Richard S. Wiener.
Data structures using Modula-2
Wiley, c1986.

Jones, William C. Jr.
Data structures using Modula-2
Wiley, c1988.

Stubbs, Daniel F. & Neil W. Webre
Data structures with abstract data types and Modula-2
Brooks/Cole Pub. Co., c1987

Feldman, Michael B.
Data structures with Modula-2
Prentice Hall, c1988.

Wiatrowski, Claude A. & Richard S. Wiener
From C to Modula-2--and back, bridging the language gap
Wiley, c1987.

Welsh, Jim & John Elder
Introduction to Modula-2
Prentice-Hall, c1987.

Chirlian, Paul M.
Introduction to Modula-2
Beaverton, Or. : Matrix Publishers, c1984.

Ural, Saim & Suzan Ural
Introduction to programming with Modula-2
Harper & Row, c1987.

Greenfield, Stuart B.
Invitation to MODULA-2
Petrocelli Books, c1985.

Schnapp, Russell L.
Macintosh graphics in Modula-2
Prentice-Hall, c1986.

Metrowerks, Inc. Staff
Metrowerks Modula-2 Start Pak.
Macmillan, 1990
(includes a disk)

Beidler, John, & Paul Jackowitz.
Modula-2
Boston : PWS Engineering and Computer Science, c1986.

Cooling, J.E.
Modula-2 for microcomputer systems
Van Nos Reinhold, 1988

Gleaves, Richard.
Modula-2 for Pascal programmers
Springer-Verlag, c1984.

Novak, M.M.
Modula-2 in Science & Engineering.
Gregg-McGraw, ?1990?

Walker, Robert D.
Modula-2 library modules: a programmer's reference
Tab Books, c1988.

Schildt, Herbert
Modula-2 made easy
Berkeley, CA : Osborne McGraw-Hill, c1986.

Kelly-Bootle, Stan.
Modula-2 primer
Howard W. Sams & Co., c1987.

Etling, Don.
Modula-2 programmer's resource book
Tab Professional and Reference Books, c1988.

Ogilvie, John W. L.
Modula-2 programming
McGraw-Hill, c1985.

Kaplan, Ian & Mike Miller
Modula-2 programming
Hayden Book Co., c1986.

Knepley, Ed & Robert Platt
Modula-2 programming
Reston Pub. Co., c1985.

Walker, Billy K.
Modula-2 programming with data structures
Wadsworth Pub. Co., c1986.

Harter, Edward D.
Modula-2 Programming. A First Course.
Prentice-Hall, 1990

Sincovec, Richard F. & Richard S. Wiener.
Modula-2 software components
Wiley, c1987.

Moore, John B. & Kenneth N. McKay.
Modula-2 text and reference
Prentice-Hall, 1987.

Wiener, Richard
Modula-2 wizard's programming reference.
Wiley, c1986.

Messer, P. A. & I. Marshall
Modula-2, constructive program development
Blackwell Scientific Publications, 1986.

King, K.N.
Modula-2: a complete guide
D.C. Heath & Co, 1988.

Joyce, Edward J.
Modula-2: a seafarer's manual & shipyard guide
Addison-Wesley Pub. Co., c1985.

Gough, K. John & George M. Mohay. [Australians]
Modula-2: a second course in programming
Prentice Hall, c1988.

Ford, Gary & Richard Wiener.
Modula-2: a software development approach
Wiley, c1985.

Thalmann, Daniel.
Modula-2: an introduction
Springer-Verlag, c1985.

Sale, Arthur H. J.
Modula-2 : discipline & design
Addison-Wesley, 1986

Jones, William C. Jr.
Modula-2: problem solving and programming with style
Harper & Row, c1987.

Cooper, Doug
Oh My! Modula-2!
Norton, 1990

Woodman, Mark et al
Portable Modula-2 Programming
McGraw-Hill International (UK) 1989

Koffman, Elliot B.
Problem solving and structured programming in Modula-2
Addison-Wesley, c1988.

Eisenbach, Susan & Cristopher Sadler,
Program Design with Modula-2
Addison-Wesley, 1989

Leestma, Sanford & Larry Nyhoff
Programming & problem-solving in modula-2.
Macmillan, 1989

Sawyer, Brian & Dennis Foster.
Programming expert systems in Modula-2
Wiley, c1986.

Wirth, Niklaus.
Programming in Modula-2, 3rd, corr. ed.
Springer-Verlag, 1985.

Wirth, Niklaus.
Programming in Modula-2, 4th ed.
Springer-Verlag, 1987.

Mayer, Herbert G.
Programming in modula-2. the art & the craft.
Macmillan, 1988

Kruse, Robert L.
Programming with data structures modula-2 version.
Prentice-Hall, 1988

Budgen, David
Software development with Modula-2
Addison-Wesley, c1989.

Pomberger, Gustav.
Software engineering and modula-2
Prentice Hall, 1984.

Hewitt, Jill A. and Raymond J. Frak.
Software engineering in Modula-2 : an object-oriented
approach
London : Macmillan, 1989.

Wiener, Richard & Richard Sincovec
Software engineering with Modula-2 and Ada
Wiley, c1984.

Lins, C. (Charles)
The Modula-2 software component library
New York : Springer-Verlag, c1989-
Four volumes
Source code available on disk from author.
Generic routines based on authoritative algorithms (from
Knuth, Hoare, Wirth, etc.).
Detailed bibliography. Scholarly approach.

Lins, C.(Charles)
The Modula-2 software component library volumes I-IV
Springer Verlag, 1988 - 1990

Sutherland, Robert J.
The professional programmer's guide to Modula-2
London: Pitman, 1988.

Riley, David D.
Using Modula-2: an introduction to computer science I
Boyd & Fraser Pub. Co., c1987.

Helman, Paul & Robert Veroff
Walls and mirrors : intermediate problem solving and data.
Modula-2
Benjamin Cummings Pub. Co., 1988

USUS 1991 Election

Candidates for USUS Board of Directors

Felix E. Bearden

After having served as the administrator during the past year, I feel I have gained insight into the USUS organization, its future, and its problems. In the organization I have found and worked with the officers and directors who are all dedicated to making USUS a success. All believe, as I do, that the future is bright. The problems that we dealt with last year mainly were concerned with the transition from a UCSD Pascal users group to a more general portable language users group.

One problem that we haven't solved is our declining membership. As administrator I have had to keep up with the membership and am aware of renewals and new members. I also think I have some idea about how to increase the rate of renewals and get new members. I thank the current members of the board for the steps they have taken, namely the ads in "Journal of Pascal, Modula 2, and Ada" and "Computer Language" magazines and the adoption of "Computer Language" as a member benefit, to make the recruitment of new members easier.

If you choose me to serve as a Director, my main goal would be to increase membership in USUS. Our future is bright and I would be honored to serve with the directors who have worked so hard to preserve our organization.

Keith R. Frederick

Currently, I'm a Senior in Economics at the University of Washington and have been the USUS Secretary of the Board for the past year. Now, I seek to be on the Board of Directors and ask for your vote.

I've been an Apple II user for seven years and have been involved in the Apple II world as a volunteer Apple IIGS consultant for TechAlliance (formerly Call A.P.P.L.E.) and as a paid consultant for firms such as Elfstone Software, Edmark Corporation, Hyrax, and Communitech. While I am involved with Apple II's and will continue to be, I'm also a new PC (a '386) owner and will be using that as my primary development tool in the future.

As a Board Member, I will seek to maintain (and further) support for those using the P-System as well as progressively work towards making USUS a major resource for programmers (academics, household-hackers, and professionals) of all platforms who use Pascal, Modula-2, or Ada.

Gary R. Gibb

I've been involved with the p-System since attending UCSD in the mid seventies. I was a contract programmer for Network Consulting Inc., where I worked on the Firefox editor (an ASE look-alike, with support for DOS and p-System text files) and ASF (a point&shoot DOS/p-System filer). I currently own and operate a small software company which specializes in hotel industry programs.

As a director of USUS, I would direct our efforts towards convincing the University of California to place the entire p-System in the public domain. This course of action would allow the release of creative energies, and ideas, which would help propel the p-System into the future.

A. Robert Spitzer

Statement not received by the publication deadline.

Continued on Page 33

Board Meeting Minutes (June 27, 1990)

By Keith R. Frederick

Minutes of the Board Meeting of USUS, Inc., held in room 1 of the MUSUS forum teleconferencing facility on the CompuServe Information Service June 27, 1990.

Present at the meeting were:

| <u>User ID</u> | <u>Name</u> |
|----------------|------------------------|
| 71016,1203 | Stephen Pickett (sfbp) |
| 72767,622 | Tom Cattrall (TomC) |
| 74076,1715 | Felix Bearden (felix) |
| 73007,173 | William Smith (Wm) |
| 76702,513 | Harry Baya (Harry) |
| 72257,1162 | David R. Babb (Babb) |
| 75226,3643 | Bob Spitzer (BobS) |
| 71735,1776 | Sam'l Bassett (Samlb) |

William Smith called the Board Meeting to order.

Topics discussed were:

I. Bylaw Changes

Bob Spitzer started the conversation by suggesting that any Board Of Director not attending 50% of all meetings without excused absence should not be allowed to run for reelection. William queried the Board if they read Felix Bearden's message about the proposed changes to the bylaws. Stephen Pickett indicated he hadn't but Bob Spitzer, Tom Cattrall, and Harry Baya replied that they had read it.

Stephen stated that he agreed with Bob but questioned if the Board can constitutionally do the amendment and how must it be ratified. Felix mentioned a small substitution (using "three" in place of "x" and "four" in place of "y") in his proposed change, indicating that this change was a board action and not his. Felix also commented that the second purpose of the bylaw change is to assure that the board can continue to act if members, willfully or not, stop attending meetings.

Tom Cattrall also agreed with Bob Spitzer's proposal but also stated that it didn't address the full problem of not having a quorum for meetings. Tom said he didn't really care what approach is used to solve the problem and that any solution that works would be fine.

Bob Spitzer mentioned that, if summer schedules are allowed, about ten meetings a year would take place and that missing five might be reasonable but not more without good reason. Also, Bob noted that under the other proposal, by the time one misses four meetings and misses the reinstatement period and the mechanism is put in place to elect a substitute, almost a year would have past. Thus, Bob said, the proposal should be modified to tie it to reelection.

Stephen Pickett asserted that he would like to point out the fact that the board always functioned perfectly on two meetings a year until recently. Bob Spitzer and Tom Cattrall agreed. Tom also commented on Bob's previous statement by stating that "the advantage of removing non-performing BOD members is that it's a lot easier to get >=50% of 4 than it is to get >=50% of 5 when you have members that never show up."

Bob Spitzer commented that preventing non-performers from running for reelection would weed the offenders out in one pass. However, Bob continued to say, that the other issue having to do with excessive hot air at some meetings and excessive meetings, was addressed a year or two ago with proposal he had made. The proposal was that no issues would be discussed at meetings unless a formal motion be drafted, posted in advance, and opened for possible review and discussion prior to the meeting.

Stephen asked David Babb if he had any opinion on the general constitutionality of the board passing such a bylaw without a full meeting of the membership.

David Babb indicated that passing the change would be going over the limit of the rules but also questioned who really is interested other than the few who are present. Also, David stated that the USUS meetings get nowhere because it takes so long to get things decided. David ended by saying that to do things right there should be a clear-cut proposal to be voted on.

Felix stated that, according to the bylaws, the amendment must be taken to the membership. However, Felix also said that any member can propose changes, but in this case it was the board that originally discussed doing something.

Tom Cattrall agreed, saying that the board is not trying to pass the amendment themselves, but rather come up with a proposal that is submitted to the USUS membership. Bob Spitzer suggested that Felix and himself draft a streamlined proposal which will be posted and be in the newsletter as a proposal to the membership. Felix moved to table the bylaw changes pending Bob Spitzer's and his re-submitting of a new proposal. William Smith called for a vote.

Tom Cattrall, Bob Spitzer, and Felix Bearden voted yes.

After a comment by Stephen Pickett questioning how the board can table something which wasn't a motion, the board moved on to other business.

II. JPAM Advertisement

Bob Spitzer started by moving to accept Felix's ad, with any minor amendments by Felix, Henry, and Tom be submitted as soon as possible.

William called for a discussion.

Tom Cattrall stated he thought the ad was OK and had shown it to others who liked it as well. However, Tom questioned that if the La Jolla Box address is changed, the advertisement should also have the new address. Bob Spitzer pointed out that his original motion allowed for small changes without reference back to the BoD for approval.

William Smith called for a vote.

Bob Spitzer, Stephen Pickett, Tom Cattrall, William Smith, Felix Bearden, and Harry Baya all voted yes and the motion was passed.

III. USUS Library

Tom Cattrall said since he didn't get a chance to look at the library submission form that the discussion be postponed until the next meeting. So, the discussion was postponed.

IV. USUS P.O. Box and Mail

William Smith initiated by stating the problem was that while in the past two or three years, since he started getting the mail, he was going to San Diego three or four times a week and could easily pick up the mail. However, in the last few months he has been only going to San Diego once every two or three weeks. Thus, William suggested the following options.

1. Keep the current address and
 - A. Find a USUS member in the area to take over the job.
 - B. Hire someone to forward all the mail to the adminis-

trator.

- 2. Change the USUS address to that of
 - A. the administrator
 - B. local to himself (William Smith)
 - C. a different location

William stated that:

1A is the best solution but wasn't aware of anyone in the area;
 1B involves a higher cost and more work;
 2A means the address changes with the administrator; and
 2B means he would keep doing the job and the costs would be the same.

William then moved to adopt solution 2B. Tom Cattrall agreed with William as long as the commitment is long term. Also, Tom commented that he didn't like the idea of having the address changing with the administrator.

Stephen Pickett emphatically said STOP. Stephen reminded everyone that you can't forward mail from one PO Box to another and that USUS went through the whole thing of moving the box during the reign of Pecan and it almost cost us the society. Stephen indicated that he preferred to try and find someone to handle the mail without moving the box.

Bob Spitzer said he thought that checking the box once a month would be fine; indicating that he couldn't imagine business so pressing that a couple of weeks lag could make a difference. Bob also stated being leery about moving the box at a time when membership is low.

William stated that he couldn't guarantee getting to the P.O box once a month. David Babb commented that one could pay postage ahead and have the postmaster forward mail by express mail to the new address periodically. The cost being nine or twelve dollars.

Stephen, having just looked through the roster of members, stated that he had found three members by the time he got to 'E' that live in San Diego, in fact, one in La Jolla. Stephen suggested something

could be arranged with one of them.

Felix indicated he liked David's idea. Bob Spitzer suggested that USUS put in place David's contingency plan to forward mail and in parallel, an attempt to locate someone who would pick up the mail, as Stephen wanted.

Tom Cattrall said his first choice would be to have a person get the mail at the La Jolla box because it would cut delays in getting the mail to the right USUS official. Bob Spitzer requested that the initial motion to move the box be voted down and a new motion that would provide the two level approach be formulated. So, Bob Spitzer, Stephen Pickett, and Tom Cattrall voted no to the initial motion.

Bob Spitzer then moved to redirect mail unless a volunteer is located to pick up the mail and that regardless of the method, the same box is used.

William called for additional discussion before the vote, there was none.

Stephen Pickett, Felix Bearden, Bob Spitzer, Tom Cattrall, and Harry Baya voted yes. William Smith abstained. The vote passed.

V. Miscellaneous

Stephen Pickett made a call for any help/ideas/material as he is making a real effort to organize the MUSUS material and some of his own stuff into library disks. Stephen also mentioned he is planning on putting something in the newsletter.

NEXT MEETING

The Board adjourned and agreed to meet again at 6:30 PM PST / 7:30 MST / 8:30 CST / 9:30 EST August 8th, 1990 in Room 1 of the MUSUS conference facility.

Minutes submitted by: Keith R. Frederick

Board Meeting Minutes (August 8, 1990)

By Keith R. Frederick

Minutes of the Board Meeting of USUS, Inc., held in room 1 of the MUSUS forum teleconferencing facility on the CompuServe Information Service August 8, 1990.

Present at the meeting were:

| User ID | Name |
|------------|--------------------------|
| 71016,1203 | Stephen Pickett (sfbp) |
| 72767,622 | Tom Cattrall (TomC) |
| 74076,1715 | Felix Bearden (felix) |
| 75226,3643 | Bob Spitzer (BobS) |
| 73760,3521 | Keith Frederick (KeithF) |
| 71515,447 | Alex Kleider (AlexK) |
| 71735,1776 | Sam'I Basset (SamIb) |

Bob Spitzer called the Board Meeting to order.

Topics discussed were:

I. Changes to the bylaws

Bob Spitzer started by saying that he takes Felix Bearden's latest recommendations as a potential motion and asked for a discussion. Stephen Pickett stated that he had one minor amendment: that the words "more than" be deleted from section VIII.2. Stephen said

that attendance at half of the meetings would be sufficient.

Tom Cattrall said he would vote for either the auto removal proposal or the inactive members proposal. Indicating that the other proposal doesn't address the problem.

Stephen further clarified his statement about deleting "more than" from Felix's proposal. Stephen illustrated that the proposal deals with MORE than half and LESS than half situations, but does not address the case of exactly HALF. Stephen suggested that the proposal requirement be set to HALF or more of the meetings and that removal be triggered, by whatever mechanism, by missing MORE than half of the meetings.

Bob Spitzer replied that Felix also has a proposal that reverts to a simple removal for missing N meetings. Also, Bob noted that this proposal is closer to the original proposals. However, Bob also commented that the proposal does not have a provision for notifying a Board Member that they had been removed and for them to respond.

Tom Cattrall then moved to adopt the proposal made by Felix in MUSUS message 65326 with the amendment that at N-1 meetings missed, the member be notified by mail and if the N'th meeting is

missed then the removal is automatic.

Stephen Pickett indicated that he liked the ARS proposal much better.

Bob Spitzer asked for a second and also stated that the motion must actually be "to draft the amendments and bring them to a vote of the members." Felix asked if anyone filled in the numbers in the proposal and also stated that he didn't think it necessary to set policy in the bylaws, but instead just provide the mechanism. Sam'l Bassett noted that most of the bylaws can be changed by a simple vote of the board but the only one that requires member approval was removing a Director.

Tom Cattrall indicated that he meant to include the parameters in Felix's message to be used in his previous motion. Tom also stated that he didn't feel too strongly about the notification part and would be willing to withdraw it.

Bob Spitzer made it clear that he felt strongly about the notification part, more so than other parts, since that was the "fair" part of the motion. Also, Bob questioned Sam'l Bassett if this proposal would require member approval since it provided a mechanism for the removal of a Director. Sam'l Bassett replied that even though he hadn't read the bylaws lately, he thought it would require a vote. Sam also stated that "fair" notification would avoid any hard feelings.

Tom Cattrall agreed that the notification is the fair thing to do. Also, he noted, after looking at the bylaws, that the board can amend anything except for changing the number of authorized members. Tom then stated that the current proposal doesn't exactly change the number but it comes close to the subject.

Bob Spitzer then asked if the board should vote to make the change.

Stephen Pickett questioned if there is an amendment. He then moved that the board draft the amendment for the membership's approval. Stephen indicated that the tone of the bylaws is clear in its intent.

Bob Spitzer pointed out that Tom Cattrall made the original motion which has not been voted on. Bob then asked if the board could get a consensus to go to the membership and then vote on it that way.

Sam'l Bassett, Tom Cattrall, and Stephen Pickett indicated they wouldn't object to going to the members. Felix Bearden said he didn't want to go to the members unless it's necessary. Bob Spitzer indicated that given the urgency of things, he would prefer to make the change without going to the members.

After some discussion whether the topic must be brought to the members, Sam'l Bassett stated that the board should vote to submit the issue to the membership, assuming Tom Cattrall yielded his original motion, and then the board can continue with other business.

Tom Cattrall withdrew his original motion. Bob Spitzer then said the consensus is that the board should or must go to the members, so the motion is to accept Felix's proposal with a "notification" modification, and then take it to the members. Bob Spitzer then called for a vote.

Sam'l Bassett, Tom Cattrall, Stephen Pickett, Felix Bearden, and Alex Kleider all voted in favor and the motion passed.

II. Offer of resignation by Alex Kleider

Bob Spitzer called for a discussion. Alex Kleider said that since he hasn't been doing much he would offer to resign. Bob Spitzer indicated that Alex was actually doing more than average and didn't see any need for Alex to resign...also indicating that an election would be a headache at the moment. Tom Cattrall, Sam'l Bassett, Felix Bearden and Stephen Pickett all agreed, indicating that Alex should

carry on and try to do whatever his time schedule permits.

Alex agreed to stay on and do what he can and noted that he just wanted to give the Board the option of replacing him.

III. JPAM Advertisement

Felix Bearden said the advertisement was finished and would be in the September/November issue of the Journal of Pascal, Ada, and Modula-2. Felix said he didn't have the exact figures on the cost of the ad but the charges won't be completed until the agency is billed by JPAM.

IV. La Jolla Mail Handler

Stephen Pickett said, with pleasure, that he had found an alternate mail handler: Casey Blank. Stephen said Casey lives in San Diego and has been a USUS member for eight years.

Bob Spitzer told Stephen to get in touch with William Smith so that he can turn over the responsibility and that all the motions and approvals regarding the decision were passed last meeting.

Tom Cattrall agreed with Bob Spitzer, stating that unless someone had some objection with the person that the Board should go with it.

V. Status of MUSUS

Tom Cattrall started with saying that the three (now four) SysOps of MUSUS haven't been putting too much time into running it and the messages and uploads go unanswered for long periods of time. Tom asked for suggestions on getting more effort into running MUSUS.

Bob Spitzer said he hoped that the Modula stuff will kick-in and help. Tom Cattrall agreed saying that the Modula-2 traffic will be what MUSUS needs. However, Tom also commented that without SysOps to merge uploads and perform other duties, MUSUS is sure to die.

Felix then noted that the company that bought out Logitech asked for information on a corporate membership with USUS. Felix said that this might help out as well. Bob Spitzer continued saying that he was hoping for new Modula-2 blood to take an active interest and that MUSUS would get rid of old dead weight SysOps.

Tom also said that Felix's news on Logitech could provide a good lead if USUS could get some leaflets into their software packages in order to acquire some publicity. Bob Spitzer offered to give Logitech a section and then to see what happens with the ad in JPAM and then what Logitech says and then make a decision in October.

Tom Cattrall said we wasn't sure what to propose but wanted to point out that there is a problem with running MUSUS and that if the situation is ignored MUSUS will disappear soon. Tom then proposed to follow up on the Logitech idea as soon as possible.

Bob Spitzer listed his specific proposals for the situation:

1. Continue to search for new members (JPAM ad is one method)
2. Follow up on Logitech.
3. Tell CompuServe what is being done to improve things on MUSUS.

Tom stated that, according to a CIS file in the Practice Forum, MUSUS is dead last of all forums in message traffic. Tom indicated his agreement with Bob Spitzer on telling CIS what is being done to improve MUSUS.

Bob Spitzer indicated that he would call CIS and he and Felix would get in contact with Multiscope, the company that bought Logitech.

Stephen Pickett moved for adjournment.

NEXT MEETING

The Board adjourned at 8:39 PM PST and agreed to meet again at 6:30 PM PST / 7:30 MST / 8:30 CST / 9:30 EST September 12th,

1990 in Room 1 of the MUSUS conference facility.

Minutes submitted by: Keith R. Frederick

Board Meeting Minutes (September 12, 1990)

By Keith R. Frederick

Minutes of the Board Meeting of USUS, Inc., held in room 1 of the MUSUS forum teleconferencing facility on the CompuServe Information Service September 12, 1990.

Present at the meeting were:

| User | ID Name |
|------------|--------------------------|
| 71016,1203 | Stephen Pickett (sfbp) |
| 72767,622 | Tom Catrall (TomC) |
| 74076,1715 | Felix Bearden (felix) |
| 73760,3521 | Keith Frederick (KeithF) |
| 76702,513 | Harry Baya (Harry) |
| 73447,2754 | Henry Baumgarten (Henry) |
| 72747,3126 | Bob Clark (BobC) |

Henry Baumgarten started the board meeting.

Topics discussed were:

I. MUSUS

Harry Baya started off by saying that CompuServe was likely to close down MUSUS at the end of the month unless a good reason is shown to them for not shutting down MUSUS. Also, Harry went on to say, that even if MUSUS is closed down, CompuServe would probably give MUSUS a section in another forum.

Stephen Pickett indicated he thought that the message volume was up and asked for some statistics on the message volume. Tom Catrall, after discussing the subject with Harry Baya and Felix Bearden, stated that the Board needs to present a proposal of what is going to be done to make the forum more successful. Also, Tom said that Felix had uploaded a list of ideas with which he agreed.

Tom went on to say that MUSUS has a chance of being the center for Modula-3 information, pointing out that Sam Harbison was quite interested in having Modula-3 information in MUSUS and that he was going to see about getting permission to upload Modula-3 material. Finally, Tom stated that the MUSUS SysOps aren't present at all anymore and Harry doesn't have the time to carry the load himself. Tom ended by asking for ideas on getting some fresh blood to help out.

Harry, responding to Stephen's earlier remark, stated that while message volume is up a little, the numbers are so small that CIS is not impressed and that MUSUS has to give them some reason to think it will grow a lot. Harry then stated that he agreed with Tom's comments.

Tom asked if anyone knew of someone who could be enlisted as a SysOp and specifically asked Harry if the SysOp Forum might be of some help. Harry answered saying that the SysOp Forum can't help much to bring this forum back to life but he would give it a shot anyway. Harry continued to say that the main SysOp duty where help is needed is in increasing the message base. The other duties are nuisances that really don't affect the usage of the forum.

Tom then asked about any news on corporate members and forum sections. Felix said there was nothing yet but sees new owners of Stride Modula-2 as good prospects.

II. \$33 Problem with JPAM

Felix started by saying that subscriptions are \$29 and that there is an additional \$33 for foreign air mail service as of January 1990.

Stephen Pickett noted that The American Chemical Society charges about \$20 a year, or the actual excess postage on subscriptions to Canadian addresses. Also, that there is some sort of exemption (by the Government) for magazines which are part of membership; which JPAM may or may not qualify. Stephen continued to say that since the issue was not brought up before the board there seems no option but to re-debate it, with an amendment that would make JPAM optional for overseas subscribers.

Felix commented that there are 27 foreign members which amounted to \$891. Tom Catrall indicated his agreement with Stephen's amendment that would make JPAM optional and that the postal rate chart shows overseas rates from \$4 to \$10 for the packages being discussed and that Canadian rates are quite a bit less. Stephen asked the how many Canadians were in the 27 foreign members and if JPAM is going to really charge Canadians the \$33.

Tom Catrall said that \$33 is reasonable given the postage rates but still give overseas people that option of having JPAM by paying the extra \$33 or to omit for X-dollars less. After a short discussion on the structure of the subscription fee, X was guestimated to be \$10.

Henry asked for a motion. Tom moved to make JPAM optional for those that the \$33 fee applies with the value of X-dollars to be worked out.

Felix stated he would like to amend the motion to go ahead and pay for the foreign subscriptions already processed.

Stephen countered by saying that if JPAM insists USUS pay \$20 per member then USUS will be out \$540 with nothing to show for it. Stephen then posed an alternative suggestion. That JPAM send USUS all extra copies and then USUS can send them out with its own overseas newsletters and include in the existing postage surcharge which was designed to handle a little more anyway. Stephen then said over time we may have to raise overseas subscriptions if the existing postage surcharge doesn't cover it.

Felix redoubled noting that in Stephen's proposal one is trading \$33 for the Administrators or someone else's "for now" free time and that the person who would be doing this task be identified. Felix then asked Bob Clark for a count. Bob Clark answered saying that USUS needs to pay an additional \$129 for the subscribers being processed.

Stephen countered Felix's redoublement saying that his (Stephen's) simplest suggestion is to lie to JPAM about foreign members and to

reduce the overseas, Canadian, and Mexican subscriptions to the rate before Bob Spitzer got us into all of this. Stephen then said that, as International Coordinator, he would be glad to send out all of the journals in the described manner.

Tom Cattrall said that Stephen's suggestion gets into too much detail that USUS can ill afford to have stretching its resources. Thus, Tom said, he would like to return to the motion with the amendment to accept the already processed renewals at the old rate.

Tom then restated his motion: make JPAM optional for those members that are affected by the JPAM extra charge. If they want JPAM, they pay the extra themselves and if they don't want JPAM they pay less than normal (\$45 - \$15 - \$X) where X appears to be \$10 or \$20 to be decided. The subscriptions already in the pipeline to be paid by USUS.

Stephen made two points in response to the restatement: 1). under the contract with JPAM, USUS probably cannot do this so USUS would've saved \$300 and spent \$600, so better to swallow the charge. 2). The amount of work in determining preferences by overseas members by the administrator, and the resulting paperchase is probably a nightmare.

Felix agreed, indicating that it complicates the "well oiled" machine set up previously. But, Felix noted, the agreement with JPAM is nothing other than a verbal agreement which does not specify anything about what members must be reported.

Tom Cattrall then said that it seems any of the mentioned approaches can be done and that the issue is more controlled by what administrative effort is needed by each and the costs. Also, that if USUS mails JPAM and pays the postage, it would be \$8.70 air to Europe, \$4 surface per issue.

Henry then called for a vote on the motion that was moved and seconded earlier. Tom Cattrall voted in favor of it while Stephen Pickett and Felix Bearden voted against it. The motion failed.

Stephen then said he votes that someone (himself or Bob Spitzer) approach JPAM and verify that we can leave off foreign names and at the same time try to negotiate a surface rate for Canada, Mexico, and the rest of the world. Also, that if said fails, that USUS with-

hold payment of \$20 per member for foreigners and reduce the subscription by the amount we just raised it (\$10).

Henry expressed his opinion that no progress is being made as an agreement cannot be agreed upon. Then Henry said he didn't see why it would be so difficult to give foreign members the option to join at one price or to join at another price with includes JPAM. They can get JPAM for \$10 plus postage or skip and save \$43. Finally, Henry suggested that the meeting be adjourned.

Bob Clark then asked about the subscriptions in the pipeline as it had been three weeks and none of the subscriptions had been fully processed. Henry replied that for those in the pipeline, USUS will have to pay.

Tom Cattrall pointed out that postponing can be costly as the pipeline increases at \$33 per entry and that USUS should unbundle JPAM for overseas members and let them choose. And, Tom noting that this is what Stephen's latest motion proposed, seconded it. Stephen then said that it was one option depending on a phone conversation.

Felix stated that he could withhold mailing subscriptions for remaining international members. Henry and Tom Cattrall agreed. Thus, Felix moved to table while Stephen asked if the others wanted him to phone JPAM and post the results on MUSUS.

Henry noted that motion to table admits no discussion. The motion to table was passed.

Stephen asked if he should approach JPAM. Felix responded saying that he recommends Bob Spitzer or Henry Baumgarten contact them if anyone does and that more discussion is needed before we do. Felix then moved to adjourn.

NEXT MEETING

The Board adjourned and agreed to meet again at 6:30 PM PST / 7:30 MST / 8:30 CST / 9:30 EST October 10, 1990 in Room 1 of the MUSUS conference facility.

Minutes submitted by: Keith R. Frederick

Board Meeting Minutes (October 10, 1990)

By Keith R. Frederick

Minutes of the Board Meeting of USUS, Inc., held in room 1 of the MUSUS forum teleconferencing facility on the CompuServe Information Service October 10, 1990.

Present at the meeting were:

| <u>User ID</u> | <u>Name</u> |
|----------------|--------------------------|
| 73447,2754 | Henry Baumgarten (Henry) |
| 71016,1203 | Stephen Pickett (sfbp) |
| 72767,622 | Tom Cattrall (TomC) |
| 75226,3643 | Bob Spitzer (BobS) |
| 74076,1715 | Felix Bearden (felix) |

Henry Baumgarten started the board meeting.

Topics discussed were:

I. JPAM

The meeting started with a report from Stephen Pickett.

Stephen said he had called JPAM and came up with the following:

1. JPAM is actually published by SIGS Publications, Inc who bought it from John Wiley.
2. There is absolutely no flexibility in respect of the amounts charged for postage.
3. They don't mind if USUS drops overseas members from any list they have.
4. They are willing to drop ship all Canadian members copies to my (Stephen's) PO.
5. They are about to be sold (again). SIGS doesn't want JPAM, this means there may be a new owner in about 1 week.
6. I (Stephen) faxed a memo to Margarita Monk, General Manager of SIGS, saying that (points) 3 and 4 would be agreeable to us and confirming the arrangement and that she would receive a proper letter on USUS letterhead by the end of the week.
7. That it is critical to do this deal with the existing management as with any new owners all bets are off.
8. That we move to ratify my (Stephen's) actions and ensure

(point) 7.

Tom Cattrall indicated he thought that the approach was okay but that it depended on Stephen who is quite busy. Otherwise though, Tom said he would vote for it. Stephen said that he gets to his PO box every seven or ten days for his own purposes and that he shouldn't have any problems getting copies (of JPAM) mailed...provided USUS doesn't get 100 Canadian members!

Henry asked about foreign members. Stephen said drop them indicating that it is just as cheap for them to go direct to JPAM and that USUS doesn't need to surcharge them the \$10 we are socking the North American members. Tom agreed that foreign members not be part of any JPAM deal. Then Tom moved to accept the proposal. Henry then asked for a second.

Stephen said that procedurally we may need the board to give direction to one of its officers (specifically Felix) to write said letter ASAP. Then, the motion should include wording to that effect. Stephen then asked for a recap.

Tom summarized the motion:

1. All foreign members outside the US and Canada to no longer have JPAM included in USUS membership, and to not be surcharged \$10.
2. All Canadian USUS members' copies of JPAM to be sent to Stephen's PO Box where he'll remail to Canadian members (at USUS expense).
3. Felix be directed to write confirming letter to Margarita Monk by Monday.

Stephen seconded with one addition: 1b. That existing foreign subscriptions be dropped from JPAM and that they credit our accounts accordingly.

Henry called for a vote. Stephen Pickett, Bob Spitzer, and Tom Cattrall all voted yes; there were none opposed. The motion passed.

II. Submitting Bylaw Changes for Vote

Henry asked Tom Cattrall to bring everyone up to date. Tom stated that Felix was going to write something up and that, his personal feelings, USUS vote on the changes at the same time as the next BOD elections.

Henry asked Felix if had anything prepared. Felix responded that he had no more than what was originally written with additional comments and pinning down a,b,c quantities. Felix said that the bylaws he submitted were accepted.

Henry then said that the real question is whether or not something will be ready to go out as soon as there is a slate of candidates for the BOD election.

Tom indicated that although he wasn't sure of the mechanics that will be used in the elections but that ballots will probably go out sometime in mid or late December. Henry responded saying that he thought it would be a little earlier than that but that three things need to be done: find candidates, get some personal writeup from them, and to distribute the information and a ballot to the members.

Felix said that the target date be the end of November and that he would try to prepare the Bylaws revision by November 15. Felix, after looking at the bylaws, said that the BoD is elected at the annual meeting, whenever it is held. Since the meeting can be called by the BoD and there are provisions for electing BoD if the meeting is not held, the key issue is expiration of terms which is one period year from the election....in other words Felix said, indeterminate.

Tom, who was also looking at the bylaws, agreed that they are rather loose on when elections are held and thus any time we choose, as long as current outgoing board members don't object, should work. Tom indicated that the vacancies would be from Henry Baumgarten, Frank Lawyer, and Bob Spitzer.

Bob Spitzer commented though that terms were changed to two years and that BoD elections were staggered to have some continuity. Also, Bob went on to say, that there are several important issues coming up including CIS and JPAM and that elections be decided as soon as it is known where these issues are headed.

Henry replied indicating that he thought Bob was correct on his first point but that while there was no new information on the status of MUSUS, there was doubt that MUSUS would be thrown off entirely and more likely it would be placed in some other forum. Bob suggested that the board meet in two to four weeks when some of the issues will be clearer. Stephen then suggested that nominations close on November 30, postcards out by December, and voting by January 31. Also, have another meeting in three weeks.

After a lot of discourse regarding the next meeting it was agreed that the next meeting will be November 14.

III. Changed USUS Letterhead

Stephen started by saying that, from the comments on MUSUS, everyone is in favor of it. However, Stephen issued a contrary point: noting that the letterhead is something familiar, like the PO Box, he sees no reason to change it unless it accompanies a substantial change of the organization itself. In otherwards, he ended, that it should reflect change rather than be the agent thereof. Tom Cattrall said he was easy and either way (changed or not) would be okay. Henry agreed with Tom.

Stephen then had two questions for Felix: is cost an issue in the logo choice and what about the money paid to JPAM up front (in case JPAM is taken over what happens) ? Felix answered that new letterheads don't cost any more and contracts made by an owner must be honored by the purchaser.

NEXT MEETING

The Board adjourned and agreed to meet again at 6:30 PM PST / 7:30 MST / 8:30 CST / 9:30 EST November 14, 1990 in Room 1 of the MUSUS conference facility.

Minutes submitted by: Keith R. Frederick

Board Meeting Minutes (November 14, 1990)

By Keith R. Frederick

Minutes of the Board Meeting of USUS, Inc., held in room 1 of the MUSUS forum teleconferencing facility on the CompuServe Information Service November 14, 1990.

Present at the meeting were:

| <u>User</u> | <u>ID Name</u> |
|-------------|--------------------------|
| 71016,1203 | Stephen Pickett (sfbp) |
| 72767,622 | Tom Cattrall (TomC) |
| 74076,1715 | Felix Bearden (felix) |
| 73760,3521 | Keith Frederick (KeithF) |
| 73447,2754 | Henry Baumgarten (Henry) |
| 72747,3126 | Bob Clark (BobC) |
| 73007,173 | William Smith (Wm) |
| 75226,3643 | Bob Spitzer (BobS) |

Henry Baumgarten started the board meeting.

[the first 12 minutes of this meeting are not noted in this document]

Topics discussed were:

I. Elections

Henry stated that, for the elections, nominees and statements from the nominees are needed. Henry then asked for discussion of nominees for Board of Directors. Tom Cattrall nominated Felix Bearden and Bob Clark. Bob Clark replied that he has had the treasury job for nearly five years and doesn't need another job. Felix then said that since he is the Administrator he should be an employee of USUS and that may represent a problem.

Henry asked for any other suggestions, nominations, or comments. Stephen Pickett then nominated Keith Frederick. Henry asked William Smith if he had the time and interest to run. William responded, saying that he doesn't wish to be a Board Director but would volunteer his time in staff positions.

Henry asked if there were any objections to Felix being a candidate. Henry expressed his opinion saying that the conflict may be ignored assuming that it exists. Tom Cattrall agreed, saying it was okay with him. Stephen Pickett also indicated no objection to Felix being a candidate. Bob Spitzer seconded Felix Bearden and Felix Bearden then seconded Keith Frederick.

Tom Cattrall pointed out that the bylaws say that directors don't need to be USUS members and that existing board members serve until they are replaced.

Felix moved that Tom Cattrall and Stephen Pickett be appointed as a committee to put together a slate of candidates. Henry asked for a second. After no response, Felix withdrew the motion. Henry then asked for additional nominations.

Bob Spitzer commented that the work needs to be split noting that he has been a Board of Director for a while and rather tired and stretched quite thin. Also, Bob said, that he has succeeded in guiding a shift in emphasis and is please to see the increased Modula activity. Furthermore, that he is exploring getting some interest in universities where he travels and gives lectures for the ACM. Bob ended by saying that he would not run if

there is someone who wishes his spot but if there are inadequate volunteers he would run again.

William Smith asked on what kind of timeline is the Board working. Henry answered saying that the Board hasn't held to a timeline but would like to have the ballots go out in December unless there are objections. Stephen stated that perhaps the next meeting should be on the 5th of December rather than the 12th so that there would be enough time to get the ballots ready. Also, coming to that meeting with a slate of candidates.

Tom moved to put Stephen in charge and let him select any assistants he chooses in order to find another name or two for the slate. The motion passed without discussion.

II. Printing

Henry now said that printing should be considered. Henry indicated that both he and Tom Cattrall have access to a laser printer and Xerox at no cost. Tom Cattrall asked about labels. Tom also pointed out that Bob Geeslin's resignation as publisher of the newsletter should be discussed. Tom said that if there are no other ideas, he would look into getting a printer local to him and that if Felix could keep him up to date on the mailing list he could easily print the labels as needed and not burden Felix. William Smith offered to find a printer local to him that USUS could find and try it for a few issues.

Suddenly, something happened on CompuServe which caused many at the meeting to logoff and re-logon.

When the problem was over Bob Spitzer raised two issues which he said didn't have to be resolved but which are relevant to the recent success as a Modula-2 forum. The issues were: 1). What should be done to replace JPAM and 2). Is Pecan defunct. Bob asked that if Pecan is defunct can USUS approach UCSD about getting the rights to the P-System and place it entirely in the public domain under the auspices of USUS.

III. Replacing JPAM

Henry started off by saying that he was not that impressed with Computer Language's issue and asked for an alternative. Tom Cattrall replied that he didn't know of any other magazines that are devoted to Pascal and Modula-2 and suggested that perhaps dues be reduced to what they were before.

Bob Spitzer asked why JPAM hasn't replied regarding his manuscript. Tom Cattrall said he received a letter from Friedman that said Computer Language Magazine has the subscriber lists and pending manuscripts.

Felix reported on what he knew of JPAM with the following points:

1. JPAM is no more.
2. CLM may have run the USUS advertisement in the December issue.
3. CLM also asked the ad folks to return an incorrect insertion order committing them to pay \$1300+ for the third ad.
4. Bob Clark should have the letter from GrapeGraphics concerning (point 3).
5. CLM is not in line with USUS.
6. A comment to that (point 5) effect was in the letter just to get their attention.

7. It was pointed out that CLM should honor the JPAM contract.

8. The position that I (Felix) took is "let them break the contract by charging more."

Stephen responded by saying that CLM should be up USUS's streets and asked why someone doesn't talk to them personally. Also, Stephen said, that he had the impression, from the JPAM staff, that CLM should be honoring any agreements but that it might be up to USUS to point that out to the new journal staff.

Felix said he would be happy to talk with CLM after the Board tells him what direction to take: does USUS want to continue with CLM or not. Tom Cattrall added that it is his belief that CLM bought the subscriber list and that they won't change their editorial policies and that they will keep USUS as they have been both in editorial content and advertisement rates.

Bob Spitzer suggested that unless USUS can publish its own journal that CLM be provided temporarily until something better can be offered. Bob Spitzer then asked for approval to contact all student chapters of the ACM (at his own expense) to initiate contact with all universities using Modula-2 as a teaching language and make them aware of this forum. Also, submissions to the library would also be encouraged and eventually the newsletter could be enhanced by providing a forum for student papers. This, Bob said, might even eventually become a journal.

Stephen Pickett then moved Bob Spitzer's suggestion and Felix Bearden seconded. Stephen then asked if USUS will get the same break temporarily as it did with JPAM and how many issues per year is CLM. Tom Cattrall answered that CLM is 12 times per year. William Smith stated that CLM charges more for ads because it has a bigger following.

Harry commented that Bob Spitzer's idea sounded great for promoting Modula-2 on campuses. Henry called for a vote to approve Bob's idea. Bob Spitzer, Stephen Pickett, Tom Cattrall, William Smith, Felix Bearden, and Harry Baya all voted in favor with none opposing. The motion passed.

Tom Cattrall returned to the CLM topic by pointing the following three points:

1. Existing JPAM subscriptions are filled by CLM issues for same period of time means twice as many CLM issues as JPAM.
2. What to do with new USUS renewals: do they get CLM.
3. What to do about ads.

Tom said that on item 2, the Board should wait for next meeting to see what information Felix finds and on item 3 more information is needed. Henry summed it up by saying that the Board basically will just wait for Felix to get the information.

III. Elections & CLM & Printing

William Smith suggested to ask on the ballots if the members want CLM and that by that time current members would have one or two issues. Bob Spitzer, Tom Cattrall, and Felix Bearden agreed.

Stephen then issued a plan:

1. Meet December 5th (or 12th) and draft ballot.
2. Have election in February returnable February 28th.
3. Take decision on CLM around that time.

Bob Spitzer agreed but Tom Cattrall indicated that the Board shouldn't wait that long on CLM but instead take up the matter at the next meeting. Felix agreed with Tom.

Henry said that if the Board is going to solicit nominations from the members and the next newsletter goes out in December the ballots might not get out until January and asked if this was a problem. Stephen remarked that he didn't know it was decided to solicit nominations from the members. Henry replied that he inferred it from the comments.

Stephen then asked if his motion was seconded by Bob Spitzer. Bob Spitzer replied "which motion?" Bob Spitzer then said the Board should just put out a ballot since it can't even get people who are active to run. Bob Spitzer then said that if USUS does grow then it can go back to parliamentary politics and that the Board should just do something simple. Tom Cattrall and Felix Bearden agreed.

Henry asked Stephen which motion as well and Stephen clarified: have ballot in January, with opportunity for members to contact nominating committee after the newsletter in December. Then, close elections at the end of February. Stephen also pointed out that anyone can run for election via a write-in.

Henry commented that one reason he was in favor of the nominating committee is that it gives the appearance of orderly process and that he saw no real problem.

Tom Cattrall said since there is a duly voted nominating committee the Board should wait until next meeting to see how it works and now the Board should deal with the issues of printing the newsletter and the CLM issues. Tom then proposed that he and William Smith check into printing. William agreed and Felix Bearden seconded.

Tom Cattrall then proposed waiting until next meeting for CLM. Felix said "so moved." Henry stated that he thought the Board already agreed on all this and wondered why a motion was needed. Tom Cattrall replied that most agreed to it but the discussions kept going on and just wondered if a motion or vote was needed. Tom also said that he didn't think there was a vote on waiting on CLM and that a vote probably wasn't needed to have he and William look into printing.

Henry then called for a vote on waiting on CLM while Stephen Pickett moved to table all outstanding motions to the next meeting.

Bob Spitzer, Tom Cattrall, Felix Bearden, William Smith, Harry Baya, and Stephen Pickett all voted in favor with none opposing. The motion carried.

IV. Pecan

Henry asked if there was anything else. Bob Spitzer then asked if Pecan was dead and if USUS could get the rights to the P-System and put it in the public domain under USUS supervision. William Smith said "Pecan -> Liaison" and the rights probably followed.

After a brief discussion, the next meeting was agreed to be held on the 5th of December.

NEXT MEETING

The Board adjourned and agreed to meet again at 6:30 PM PST / 7:30 MST / 8:30 CST / 9:30 EST December 5, 1990 in Room 1 of the MUSUS conference facility.

Minutes submitted by: Keith R. Frederick

Treasurer's Report

by Robert E. Clark, Treasurer

July 1990

| | | |
|-----------------------------|------------|-------------|
| Bank Balance | \$6,568.61 | 6-30-90 |
| Income - July 1990 | | |
| Dues: | | (new/renew) |
| Student | 0.00 | 0/0 |
| General | 630.00 | N/A |
| Professional | 200.00 | 0/2 |
| Institutional | 0.00 | 0/0 |
| Other Income: | | |
| CIS | 26.98 | |
| Library fees | 0.00 | |
| Total Income: | \$856.98 | |
| Expenses - July 1990 | | |
| Administrator: | | |
| Shipping | 12.17 | |
| Other: | | |
| Bank charges | 2.00 | |
| Newsletter | 617.84 | |
| Mail from La Jolla | 6.05 | |
| Reimbursements | 0.00 | |
| JPAM Subs. (14) | 316.00 | |
| Total Expenses | \$954.06 | |
| Bank Balance | \$6,471.53 | 7-31-90 |

August 1990

| | | |
|-------------------------------|------------|-------------|
| Bank Balance | \$6,471.53 | 7-31-90 |
| Income - August 1990 | | |
| Dues: | | (new/renew) |
| Student | 0.00 | 0/0 |
| General | 175.00 | 0/4 |
| Professional | 0.00 | 0/0 |
| Institutional | 0.00 | 0/0 |
| Other Income: | | |
| CIS | 0.00 | |
| Library fees | 0.00 | |
| Total Income: | \$175.00 | |
| Expenses - August 1990 | | |
| Bank charges | 2.00 | |
| Newsletter | 0.00 | |
| Mail from La Jolla | 0.00 | |
| Reimbursements | 80.00 | |
| JPAM Subs. | 0.00 | |
| Total Expenses | \$ 82.00 | |
| Bank Balance | \$6,564.53 | 8-31-90 |

September / October 1990

| | | |
|--|------------|-------------|
| Bank Balance | \$6,564.53 | 8-31-90 |
| Income - September/October 1990 | | |
| Dues: | | (new/renew) |
| Student | 25.00 | 1/0 |
| General | 135.00 | 0/3 |
| Professional | 0.00 | 0/0 |
| Institutional | 0.00 | 0/0 |
| Other Income: | | |
| CIS | 62.30 | |
| Library fees | 8.58 | |
| Total Income: | \$230.88 | |
| Expenses - September/October 1990 | | |
| Bank charges | 4.00 | |
| Newsletter | 0.00 | |
| Mail from La Jolla | 1.90 | |
| Reimbursements | 25.00 | |
| JPAM Subs. | 489.00 | |
| Total Expenses | \$519.90 | |
| Bank Balance | \$6,275.51 | 10-31-90 |

November / December 1990

| | | |
|--|------------|-------------|
| Bank Balance | \$6,275.51 | 10-31-90 |
| Income - November/December 1990 | | |
| Dues: | | (new/renew) |
| Student | 0.00 | 0/0 |
| General | 760.00 | 1/13 |
| Professional | 100.00 | 0/1 |
| Institutional | 0.00 | 0/0 |
| Other Income: | | |
| CIS | 38.14 | |
| Library fees | 0.00 | |
| Total Income: | \$898.14 | |
| Expenses - November/December 1990 | | |
| Bank charges | 19.76 | |
| Newsletter | 58.38 | |
| La Jolla Box Rent | 36.00 | |
| Mail from La Jolla | 2.40 | |
| Refunds | 30.00 | |
| Reimbursements | 127.96 | |
| JPAM Adv. | 616.00 | |
| Total Expenses | \$ 890.50 | |
| Bank Balance | \$6,283.15 | 12-31-90 |

Ballot Measure I

Proposed Change to USUS Bylaws

The Board of Directors recommends to the members the following by-laws changes for the automatic removal of a board member for non-attendance at board meetings. The provisions are intended to assure that the board can continue to operate should board members not attend (or resign) for any reason.

1. provides for automatic removal of a Director for not attending board meetings. This provision removes from the membership the necessity of having to remove a Director by taking a vote.

2. provides for reinstatement of a Director who has been automatically removed, and has good reason for missing the meetings in the opinion of the then constituted board.

3. provides for establishing a quorum with vacancies. The by-laws now set the quorum based on the authorized number of directors. Should a majority of the positions become vacant for some reason. No action is possible by the Board until an election is held. This provision allows the board to operate until an election may be held to fill the positions.

1. Add to ARTICLE VIII DIRECTORS, SECTION 4 VACANCIES

(e) Automatic Removal of a Director Any director who is absent from 3 sequential regular Board meetings or a total of 4 regular Board meetings in a 6 month period shall be removed

from the Board of Directors.

2. Add to ARTICLE VIII DIRECTORS, SECTION 1 POWERS, (b) Specific Powers

(v) Reinstatement of a Director who has been removed automatically (See VIII.4(e)) provided an extenuating circumstance surrounds the removal of the director and the reinstatement occurs within 2 regular meetings of the Board.

3. Reword ARTICLE VIII DIRECTORS, SECTION 9 QUORUM, first sentence to read

A simple majority of the authorized number of Directors minus the number of vacancies shall constitute a quorum for the transaction of business, except to adjourn as provided in Section 11 of this Article.

Ballot Measure II:

Advisory on Keeping Computer Language Magazine as USUS Membership Benefit

With the demise of the Journal of Pascal, Ada, and Modula-2 (JPAM), Computer Language Magazine has taken over the JPAM subscription base. As Computer Language Magazine is not as directly concerned with the languages that are of interest to USUS members, the USUS board voted to ask the members whether they wish to have Computer Language Magazine as a benefit of USUS membership.

USUS Board of Directors ballot for the 1991 elections.

You have a total of 3 votes. You may cast one vote for each of 3 candidates, 3 votes for one candidate, or any combination in between. If you cast more than 3 votes, this ballot will be invalidated. Write the number of votes on the line beside the candidate(s) of your choice. A mark (check mark, cross, etc.) or marks on a line will count as one vote for that candidate. For the ballot measures, vote either for or against each measure.

Cut along the vertical bar on the left and fold so that the address on the opposite side of the page is facing outward. Tape (do not staple) closed. Affix the appropriate postage (25 cent stamp if mailed in the USA) and mail. The ballots must be postmarked by February 28, 1991 to be counted.

Candidates for USUS Board of Directors:

- Felix E. Bearden
- Keith R. Frederick
- Gary R. Gibb
- A. Robert Spitzer
- None of the Above

Ballot Measures:

- For Changing Bylaws as described above
- Against Changing Bylaws as described above

- For Keeping Computer Language Magazine as USUS Membership Benefit
- Against Keeping Computer Language Magazine as USUS Membership Benefit

Do not sign this ballot

**USUS
P.O. BOX 1148
LA JOLLA, CA 92038**

ADDRESS CORRECTION REQUESTED
FIRST CLASS MAIL

**Tom Cattrall
7600 Seawood Road SE
Amity, Oregon 97101**

From The Editor

by Tom Cattrall

Robert Geeslin who served so well as the person that handled the job of dealing with the printer and the postal service has decided to retire from the job. I'm sorry to see him do so but can understand the need to do other things. While learning the ropes as newsletter editor, I was very glad that Bob was there to handle the publishing duties. It's been a big enough learning experience without having to deal with printing and mailing too. So I thank him and wish him well.

William Smith, the previous editor, has offered to take over the publishing job and I'm glad to have his help.

While working on the USUS ad, the agency suggested a new logo for USUS. The result may be seen on the top of page 1.

The activity on MUSUS has increased quite a bit lately and it looks like a new core of people is starting to gather. Modula-2 activity is most prevalent, with a small amount of traffic dealing with UCSD Pascal and the P-system. Generic Pascal discussions seem to be quite rare.

A study group is just forming to study object oriented de-

sign using the book "Designing Object-Oriented Software" by Wirfs-Brock et. al. The group will have a moderator with OOD experience. The plan is to do a software design and then implement it in one or more different languages to see how different implementations using the same design turn out. The primary language will probably be JPI Modula-2. I suggested using a file archive utility as the project.

If you are interested in learning Object Oriented Design and Object Oriented Programming, or just want to help out with the project, join us on Compuserve in the Portable Programming Forum (GO CODEPORT, or GO MODULA).

There was some talk of getting a group of people together to work on porting Modula-3 to MSDOS but it looks like there won't be enough people with enough free time to undertake it right now.

It is time for the USUS elections. Besides the candidates for USUS board of directors, there are 2 ballot measures. Be sure to read the ballot statements and send in your ballot.

Submission Guidelines

Submit articles to me at the address shown on the back cover. Electronic mail is probably best, disks next best, and paper copy is last. If your article has figures or diagrams, I can use encapsulated Postscript files in any of the disk formats listed below. If you can't produce encapsulated Postscript, then paper copy is probably the only practical method for submitting graphics.

You can send E-Mail to my Compuserve ID: 72767,622, or indirectly from internet: 72767.622@compuserve.com. For disks, I can read Sage/Stride/Pinnacle format disks. Also, any MS-DOS 5.25 or 3.5 disks, and 3.5" Amiga disks. If anyone wants to send Mac format disks I could probably get someone to translate them into something I can use. Whatever you send, please mark on the disk what format it is. That will save me a lot of guesswork.

Text should be plain ascii rather than a word processor file. It

can have carriage returns at the end of all lines or only at the ends of paragraphs. What you send doesn't have to look pretty. I will take care of that. My spelling checker will take care of spelling errors too. If you want special formatting use the following conventions:

1. Underline, put an underline character at each end of the section to underline.
2. ***Bold***, put a star at each end of the section to **bold**.
3. ^*Italics*^, put a caret at each end of the section to be set in *italics*.
4. ??Special requests??. such as ??box next paragraph?? should be surrounded with "?? ??".

NewsLetter Editor : Tom Cattrall
Amity Software Inc.
7600 Seawood Road SE
Amity, Oregon 97101
503/835-1613
CompuServe : 72767,622
Internet : 72767.622@compuserve.com
tomc@techbook.com

NewsLetter Publisher : William Smith

USUS Board of Directors

| | |
|-------------------|------------|
| Henry Baumgarten | 73447,2754 |
| Tom Cattrall | 72767,622 |
| Frank Lawyer | 72401,1417 |
| Stephen Pickett | 71016,1203 |
| A. Robert Spitzer | 75226,3643 |

USUS Officers

| | | |
|------------|-----------------|------------|
| President: | Alex Kleider | 71515,447 |
| Treasurer: | Bob Clark | 72747,3126 |
| Secretary: | Keith Frederick | 73760,3521 |

USUS Staff

| | | |
|----------------|---------------|------------|
| Administrator: | Felix Bearden | 74076,1715 |
| Legal Advisor: | David R. Babb | 72257,1162 |
| MUSUS Sysop: | Harry Baya | 76702,513 |

**USUS
P.O. BOX 1148
LA JOLLA, CA 92038**

**ADDRESS CORRECTION REQUESTED
FIRST CLASS MAIL**

USUS Membership Information

| | |
|-------------------------|---------------|
| Student Membership | \$ 30 / year |
| Regular Membership | \$ 45 / year |
| Professional Membership | \$ 100 / year |

\$15 special handling outside USA, Canada, and Mexico.

Write to the La Jolla address to obtain a membership form.

NewsLetter Publication Dates

| <u>Issue</u> | <u>Due Date For All Newsletter Material</u> |
|--------------|---|
|--------------|---|

| | |
|----------------|-------------------|
| Mar / Apr 1991 | March 1, 1991 |
| May / Jun 1991 | May 1, 1991 |
| Jul / Aug 1991 | July 1, 1991 |
| Sep / Oct 1991 | September 1, 1991 |

