“... the information we possess often has nothing to do with the information we need. It has to do with how the information is packaged and presented to us.”

From *Stats*, by Bill James
A Visit from a User

• No understanding of the DDL discussion
• Overwhelmed by the size and complexity of the mmCIF dictionary
• Not confident that my software will solve their problem.
• Does not have time nor staff to devote to "serious" programming projects - seat of the pants operations
Why CIF does not work with Unix tools

- Line orientation of Unix tools
  - grep, (g)awk, sed, perl
- Field orientation of Unix tools
  - (g)awk, perl, sort
- Position orientation of Unix tools
  - diff, head, tail

- These are all piping tools - very different from many being developed for CIF.
Which leads to ... ZINC

- A piping format
- block <\t> name <\t> index <\t> value <\t> loop-id
  - new-lines replaced by "\n"
  - comments are included
- This format is accessible to most Unix tools
  (long lines are sometimes a problem with the older tools)
Applications

• zincGrep - search a CIF for a regexp
• cifZinc - convert a CIF to a ZINC
• zincCif - convert a ZINC to a CIF
• zincNI - Create a namelist input from a ZINC
• cifdiff - find real differences in CIFs
• zincSubset - Extract a subset of a CIF.
• zb - A simple browser in tcl/tk. << 200 lines
SimpleCif - 1

data_bigloop
   _name     "lots of points"
   _author
;
   Dave Stampf
;
   loop_
      _x  _y  _color
      0  0  red
      1  1  red
      2  4  red
      3  9  orange
      4 16  orange
      5 25  orange
   _status complete
zincGrep

```
bach 1% grep author simple1.cif
   _author
bach 2% zincGrep author simple1.cif
bigloop author ;
   \n   Dave Stampf
bach 3%
```
**cifdiff - the "similar file"**

```plaintext
data_bigloop
   _status  complete

loop_
   _y    _x    _color
   0     0     red
   1     1     red
   2     2     red
   9     3     orange
  16     4     orange
  25     5     orange

   _name  "lots of points"
   _author
   ;
   Dave Stampf
   ;
```
cifdiff - the result

bach 4% cifdiff simple1.cif simple2.cif
18c18
< bigloop y 2 4
---
> bigloop y 2 2
bach 5%
cifdiff - the program

```csh
#!/bin/csh
#
# @(#) cifdiff 1.1 9/24/94
#
# find difference in two cifs.
#

cifZinc $1 | sort -t\ +0 -1 +4 +1 -2 +2n -3 |
gawk -F\ -v OFS=\ '{print $1, $2, $3, $4}' > /tmp/$1.zinc
cifZinc $2 | sort -t\ +0 -1 +4 +1 -2 +2n -3 |
gawk -F\ -v OFS=\ '{print $1, $2, $3, $4}' > /tmp/$2.zinc
diff /tmp/$1.zinc /tmp/$2.zinc
rm /tmp/$1.zinc /tmp/$2.zinc
```
zincSubset - generating a cif subset

```
bach 1% zincSubset coords simple1.cif | zincCif

data_bigloop
    loop_ _x
        _y
        0  0
        1  1
        2  4
        3  9
        4 16
        5 25

bach 2%
```
zincSubset - the program

```
#!/bin/csh
#
# code to determine the values of the v and c switches removed
# for display purposes.

cifZinc $c $2 | egrep $v -f $1
```
zincNI - the application program

```fortran
program testnl
  C
  C     Get namelist to work.
  C
  integer x(6), y(6)
  namelist /bigloop/ x, y

  read (5,nml=bigloop)
  write(6,600) (x(j), y(j), j=1,6)
  600 format(12(1x,i12))
  stop
  end
```
zincNl - the result

```
bach 1% zincSubset coords simple1.cif | zincNl | testnl
  0  0  1  1  2  4  3  9  4  16  5  25
bach 2%
```
Gains and Losses

• +
  • Huge number of potential application programmers
  • Huge base of existing software
  • Empowers the individual consumer

• -
  • Big change in size
  • Unreadable in a different way than CIF