Open GENIE

- Open GENIE is for Neutrons
  - Guarantees *access to data* in a uniform fashion, regardless of external data formats and hardware platform.
  - Allows *algebraic manipulation* in terms of spectrum workspaces.
  - A mid-range tool for *visualising* single and multiple spectra data.
  - Open GENIE will be available as free software in the public domain with source code.
- What is Open GENIE *NOT*?
  - Not a top end commercial visualisation tool like AVS.
  - Not an all purpose visualisation tool like IDL.

Access to data

- Data is read either from raw data files or processed data files. The formats are detected *transparently*.
  eg
  >> set/file "user:[hrpd]hrp00273.raw"
  >> myspec = s(1)

- Workspaces contain all the information that is likely to be useful for that spectrum.

  — X, Y and E values

  — Units

  — Axes labels

  — Historical data

- Information may be added *dynamically*.
  eg
  >> myspec.comment="Excellent result"

- Information is then carried around with the workspace through calculations.

Contents of "myspec"

(run_no) = (00273)
(file) = (user:[genie]hrp00273.raw)
(inst_name) = (HRPD)
(run_duration)=(37781.000000)
(user_name) = (Willis/Carlile/WTAH)
(time) = (2-DEC-1986 22:44:12)
(spec_no) = (1)
(emode) = (0)
(efixed) = (0.000000)
(delta) = (0.000000)
(twotheta) = (177.889999)
(fi) = (0.000000)
(ntc) = (2422)
(x) = (Real array ... 2423 elements)
(y) = (Real array ... 2423elements)
(e) = (Real array ... 2423 elements)
(title) = (PG phi=289.8deg 20ms window)
(xlabel) = (Time-of-Flight (\gms))
(ylabel) = (Neutron counts / \gms)
(bin_grouping)=(0)
(ymax) = (693.533835)
(l1) = (96.000000)
(l2) = (2.100000)
(history) = ( )
(nut) = (1)
(ut) = (Real array ... 1 elements)
(comment) = (Excellent result)

Data Manipulation

- Open GENIE is designed to allow algebraic manipulation of neutron data.

  — Can perform calculations on:
  Integers, Reals, Arrays and Spectrum workspaces.

  — Defines high level functions useful for neutron data.
  eg
  Summing
  Units conversion
  Rebinning

- Allows the user to define custom functions/procedures which become new commands within Open GENIE.

- Procedures can use a highly flexible set of control constructs available.

  — Selection/Alternatives IF, IF-ELSE, CASE

  — Iteration LOOP, LOOP-FROM-TO, EXITIF

Example correction procedure.

An example of preliminary data analysis with a monitor correction. The procedure below averages a group of spectra then normalizes against a corrected monitor. Finally changing units to wavelength.

```
PROCEDURE Norm_sum
PARAMETERS first last
RESULT gspec
LOCAL mon
LOCAL unt=3.584e-3 sig=0.013
  mon = s(50)
  gspec = spadd(first,last) /  (last-first+1)
  mon = mon / (unt * (1- exp( -8.3 * sig * mon.x )))
  gspec =rebin(mon,gspec) /  mon
  gspec = units:lam(gspec)
ENDPROCEDURE
```

Using the procedure in an interactive session

```
>> set/file "iris$disk0:[irsmgr.data]irs09089.raw"
>> final = norm_sum(2,20)
>> alter/binning 1
>> display final[1]
```

[1] See plot 1

Displaying data

- Open GENIE provides easy to use Display routines, similar to GENIE-V2

  — These provide an easy way to get a workspace plot on the screen. eg Display, Plot, Alter.

- The basic display routines are written as procedures from a comprehensive set of graphics primitives.

  — allows customisation or re-writing for any high level commands.

- Open GENIE graphics are built using a fully Object oriented graphics library[1] coded in C++.

- The Object Graphics library adds hardcopy and display-list functionality to underlying PGPLOT primitives.

[1] Available as free software separately

Graphics Example

Customising the first plot to add colour and save in colour postscript.

```
>> alter/textcolour red
>> alter/plotcolour blue
>> alter/binning 1
>> display final[1]
>> hardcopy "plot2.cps" "cps"
```

Now adding a small insert plot of the monitor spectrum. This would normally be done in a procedure as the commands are a little fiddly to use!

```
>> mon = s(10)              # need to get this again
>> window/autoscaled 0.2 0.5 0.35 0.6 &
mon.x mon.y mon.e yellow
>> draw/axes                # already scaled above
>> alter/textcolour green
>> draw/title "Raw monitor spectrum"
>> draw/labels mon.xtitle mon.ytitle _  0.5
>> alter/plotcolour red
>> plot mon                 # plot in window
>> hardcopy "plot3.cps" "cps"[2]
```

[1] See plot 2
[2] See plot 3