West CHEM University of Glasgow Department of Chemistry

IUCr Commission on Crystallographic Computing
Siena 2005: Crystallographic Computing School

Louis Farrugia

Lecture-2

Connecting programs together

Some crystallographic programs have complex functionality – *program systems*

- PLATON, CRYSTALS
- DIRDIF, WinGX, GSAS, SHELXTL ...
- CCP4, XtalView, X-Plor, Phenix
- public SHELX suite

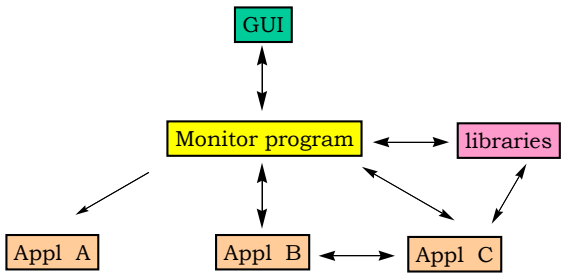
- monolithic program (single executable)
- separate executables connected together
- simply separate executables

Advantages of connected program systems

- file formats/data interconversions automatically handled
- monitor program(s) controls flow of processing by application programs
- changes easier to make - GUI/Monitor/applications can be updated separately
- coding errors *may* be less easily propogated - adding new code to monolithic programs may introduce unwanted side-effects

Disadvantages

- program interconnection *can* be dependent on operating system



Simple program system architecture

GUI

Monitor program


```
do
  if (menu1_selected)  do X
  if (menu2_selected)  do Y
  if (buttonA_pressed) do Z
  .
  .
enddo
```

Event loop

In WinGX system, executable wingx32.exe is both

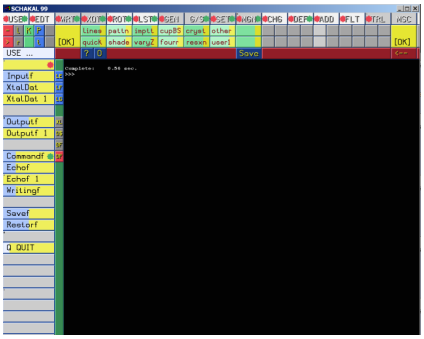
GUI design

GUI design is more an art than a science

- keep it simple – limit the number of choices on an individual dialog box
- keep operations as standard as possible
- lay out controls neatly

- keep to the "look and feel" appropriate to your platform – use native windowing tools

GUI design



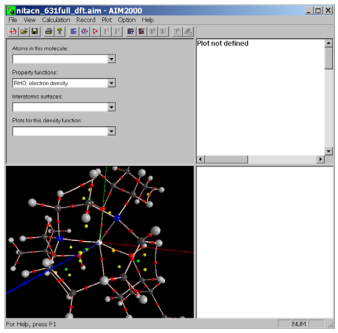
Symbols very cryptic

Need a very good understanding of the program to use GUI

Windows pop up with no obvious way of dismissing them

GUI design

Don't include features which irritate users !



- pointless dialog boxes
- use all system resources
- no useful text output

The purpose of a well-designed GUI is to make it *easier* for the user.

Libraries

All large program systems make use of libraries.

library code has general functionality and is used in several subprograms.
a = matmul(b,c) – a,b,c are arrays

library routines have well-designed interfaces - reuseability is emphasised in OO programming.

why use libraries ? efficiency in programming – no need to reinvent the wheel

Libraries

WinGX has 6 libraries implemented as DLL's

- wgxlib00.dll – mathematical functions – matrix inversion, eigenvalue, cell transformations, general routine, sorting, free-format parsing *etc*
- wgxlib01.dll – encapsulation of Salford routines
- wgxlib02.dll – PGPLOT graphics libraries
- wgxlib03.dll – GETSPEC space group routines
- wgxlib04.dll – encapsulated Salford GUI routines
- ciftbx26.dll – CITtbx version 2.6.2

Libraries

Why use dynamically linked libraries ?
(<name>.dll Windows <name>.so Linux)

- saves space – only one version needed – many executables can be linked simultaneously to same DLL

- makes correcting/updating large program systems easier

- code is linked at compile-time with library – changes can be made, but interface must remain the same

Limitation - must be self-contained set of routines - no calls to external routines.

Libraries

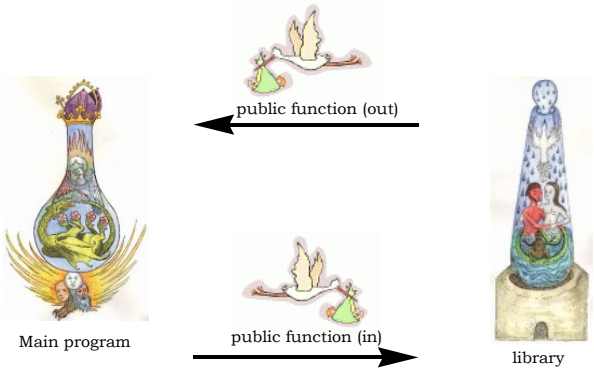
CIFtbx (modified version 2.6.2) implemented in library ciftbx26.dll

Public functions passing information to DLL

ciftbx_init()	initialise all CIFtbx variables (initially in undefined state)
ocif_(<cif name>)	load contents of existing CIF into CIFTBX memory
pcif_(<cif name>)	creates new CIF in CIFTBX memory
dict_(<dic-name>)	loads a CIF dictionary in CIFTBX memory for data validation
pchar_(<string>,<value>)	puts value of CIF data item contained in <string> into new CIF

Public functions returning information from DLL

char_(<string>,<value>)	gets value of CIF data item contained in <string>, in this case character value

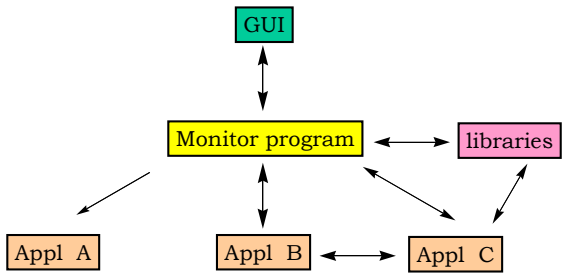


Libraries

CIFtbx version 2.6.2 implemented in ciftbx26.dll

In WinGX, the library is used to :
- import CIF's and convert to SHELX files
- validate CIF's (IUCRVAL)
- write *archive.cif*, the summary file of structure determination

last functionality involves ...
1. concatenating all CIF's into one file (data_ blocks)
2. reading request list of data items for *archive.cif*
3. sequentially finding all requested data items from concatenated CIF, and placing in *archive.cif*



Simple program system architecture

Application programs

These are separate executables – capable of being run outside program system.

How does one executable program start another ?

Highly specific to compiler/language
- Unix Fortran – call system(<string>)
- Salford Fortran – start_process@(<string1>,<string2>)
- C language – fork/exec

Scripting languages offer better solution for portability

Ousterhout's Tcl/tk scripting language is an example

Application programs

How do programs communicate with one another ?

- files – the most portable method
- operating system specific messaging

Timing of events needs to be considered

Monitor program
.....
launch program(<progname>)
read results(<filename>)
.....

Application programs

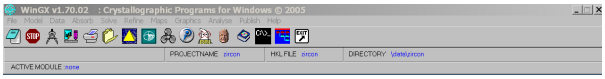
How do programs communicate with one another ?

- files – the most portable method
- operating system specific messaging

Timing of events needs to be considered

Monitor program
.....
launch program(<progname>) – does control return?
read results(<filename>)
.....

WinGX – a connected set of programs



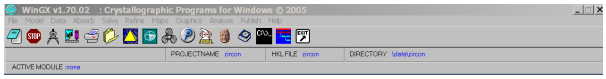
Main program is WinGX32.exe – GUI + monitor program + file-handlers +....

/bin/ ~ 70 separate executables
/files/ ~ 50 system files
/manuals/ ~ 25 pdf files

Interfaces to external programs – SirWare, CCDC programs, POV-Ray, CRYSTALS, JANA2000

Louis Farrugia - IUCr Computing School - Siena 2005 - Connecting Programs Together

WinGX – a connected set of programs



When WinGX main program is started ...

- initialises system variables and checks Windows state
- reads INI file -> current project name & location
- loads the current structural model into memory from SHELX files.
- executes code for GUI main Window, which enters event loop

Louis Farrugia - IUCr Computing School - Siena 2005 - Connecting Programs Together

WinGX – a connected set of programs

```
call wgxWindowCaptionW(MenuCaption)
call wgxWindowSizeW(MenuWindowDepth,MenuWindowHeight)
call wgxSetPositionW(0,0)
.
.
call wgxAddMenuItemW(1,'&Refine[SHELXL-97]',1,fmenu6,0)
call wgxAddMenuItemW(2,'Set HKL File',1,fmenu6,0)
.
.
call wgxDefineClassNameW('WINGXMAIN')
call wgxDefineMessageCallbackW(message_proc)
call wgxCloseControlW(exit_proc)
.
call wgxCloseDialogBoxW(0,i,0)
end
```

Louis Farrugia - IUCr Computing School - Siena 2005 - Connecting Programs Together

WinGX – a connected set of programs

```
call wgxAddMenuItemW(1,'&Refine[SHELXL-97]',1,fmenu6,0)
call wgxAddMenuItemW(2,'Set HKL File',1,fmenu6,0)
call wgxAddMenuItemW(2,'|,Open SHELXL.LST',1,fmenu6,0)
call wgxAddMenuItemW(2,'Open INS File',1,fmenu6,0)
call wgxAddMenuItemW(2,'Open RES File',1,fmenu6,0)
if(SysExec(14) /= ' ') then
  call wgxAddMenuItemW(2,'|,CRYSTALS',1,fmenu6,0)
endif
```