

CCP4i and Automation: Opportunities and Limitations

Peter Briggs, CCP4

Introduction: where does CCP4i fit into Automation?

From notes of the June 2005 meeting:

- CCP4i “is a key element of automation”
- “... it is a crucial CCP4 flagship feature.”
- “This is the public face of CCP4”
- “... would like a set of standard tools for constructing a GUI”

Access to automation via CCP4i is clearly necessary

What are the key issues that need to be addressed?

- Limitations of CCP4i
- Requirements from automation
- Challenges for meeting requirements
- How to move forward

Limitations of CCP4i wrt Automation

Operates in “batch mode” paradigm

- user sets up task, runs and reviews at the end
- poorly suited for interactive and “semi-interactive” modes

Presentation of task output is particularly weak

- general weakness of CCP4i

Job history handling is insufficiently sophisticated

- being addressed via BIOXHIT db work – won't talk about this here

Data transfer between tasks/“pipelining” of tasks

- pilot project in 2002 with transferring data for MR
- no infrastructure for doing this at present
- data transfer may also be facilitated by BIOXHIT db work

CCP4i for Automation: challenges

There are two main challenges:

- Making a CCP4i toolkit suitable for interfaces to automated tasks
- Improving the tools available to present output to the user

What are the requirements for a toolkit?

From June 2005 meeting notes:

“Tasks need to be invoked from both a GUI and via the command line.
They are required first to succeed, only secondary importance to report progress to user.”

This reflects the current CCP4i mode of operation to 1st approximation

Specifically:

1. Task interface sits on top of tasks that are invoked from the command line
2. There is minimal reporting of task outputs to the user during the run
3. Batch mode means there is minimal interaction between user and running task (non-interactive)

Note also current CCP4i can already run Python scripts from Tcl/Tk interfaces

More sophisticated interfaces

There is no specification or requirements list yet

- it would be useful to gather this information

Semi-interactive model is one possibly

- user starts task, can monitor progress outputs and interact with task as it runs
- example is HKL2MAP (Thomas Schneider/Thomas Pape's SHELX pipeline interface)

This model depends on the requirements of the automated tasks, e.g.

- are there well-defined “break points” where tasks can be interrupted and restarted?
- how long do the tasks take to run? Is it realistic to expect the user to monitor them in real time?
- what level of control should the user be allowed to exercise over the task?

An aside: technology choices

Limitation of the current CCP4i is that graphical interfaces must be written in Tcl/Tk

- personally I don't think that this is such a bad choice
- however current CCP4i architecture means that programmers have no choice
- could we enable interfaces to be developed in any language and still work with CCP4i?
- a number of toolkits could be developed independently, or programmers could experiment with third party toolkits

Architecture limitations: “Does CCP4i need rewriting?”

My answer: yes (but not from scratch)

Migration to an “open architecture” is preferable

- open architecture = modular, separates graphical and non-graphical components
- centre on a CCP4i server process (aka “resource manager”) – similar concept to database handler
- socket communications between applications, interfaces and server could be made language neutral
- new interfaces could be developed in any language (in principle) and then plugged into CCP4i easily
- existing codebase and tasks could still be used
- would facilitate development of pipelining within CCP4i
- work on the handler will give us a head start on the practicalities

Better monitoring tools: smart output viewer

Monitoring tools = anything that presents task output to the user

Current CCP4i flat logfile presentation is already insufficient for existing tasks

- a “smart” viewer would be better
- also useful for “conventional” CCP4i

Desirable characteristics for a smart viewer (wish list):

- filter content of logfile and other output to present only the most relevant details, and then allow “drill down” to more details if desired
- able to show graphs and tables in-line with text
- link to output files and allow launching of appropriate viewers
- offer some analysis of key results and link to appropriate documentation or other help resources
- allow interaction with a running task, or restart from different points if task has finished (link up with GUI toolkit)
- update in real time

How do we develop such a tool?

Possibilities include:

- Making use of XML output from programs and pipelines (“inside out” approach)
 - Investigate use of technologies such as stylesheets and XSLT as well as custom tools
- Build a library of functions to extract key data from logfiles (“outside in” approach)
 - First version could be a collection of regular expressions plus a log file parser
 - Develop into a logfile DOM (document object model) similar to that for HTML documents
- A hybrid approach combining aspects of both

Limitations on resources

Q. Who will do the work?

- It's not clear that there is spare capacity (i.e. people and time) to devote to any of these tasks
- From June 2006 meeting notes:
 - “Everything takes longer than expected”
 - “Volunteer contributors have other responsibilities; they tend to work hard for a spell, then turn to other tasks. This is difficult for the professional CCP4 staff”
- Could the STAB set priorities and assign (or lobby for) resources if they consider this work to be sufficiently important
- Adoption of open CCP4i architecture opens the door for ad hoc developments (not ideal but might get the job started)

Summary

- There is a need to investigate the needs for a GUI toolkit for automation, and to determine whether current CCP4i tools are sufficient
- Investigate development of smart browser by building on XML output plus tools for parsing logfiles and rendering output
- Investigate reimplementation of core CCP4i into open architecture model
- Requirement to assign priorities and resources to each of these projects

