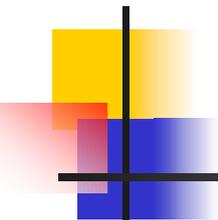


The ^{my}Grid Project

Problems and Solutions so far...

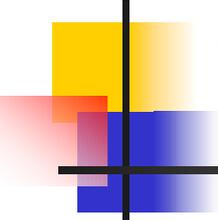
Tom Oinn, tmo@ebi.ac.uk

www.mygrid.org.uk



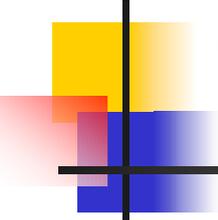
...but first, some history

- Bioinformatics is something like twenty five years old.
- Can be split roughly into 'eras', where different ways of doing the job were prevalent and particular problems present themselves.



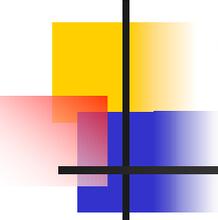
The era of hacking

- The dawn of bioinformatics.
- No one had really written any tools, so you had to write your own.
- Similarly for data formats.
- Everything in plain text, designed to be human readable.
- Few users, and they had to be experts, but no real confusion.



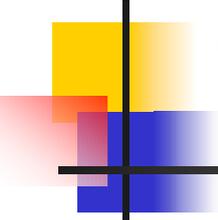
The bioinformatics tool age

- Tool sets such as GCG and Staden appear.
- Tool reuse, users run tools from UNIX command lines.
- Advanced users write shell scripts, mostly in PERL.
- Everything is still in plain text, with little standardisation.
- Some less expert users appear, get confused by UNIX shells, recover, do science.



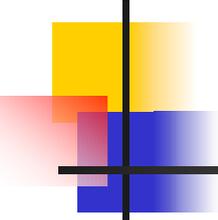
The web age

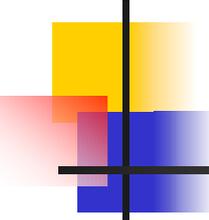
- Magic of the Internet[tm] hits bioinformatics.
- Simple CGI interfaces start appearing to existing tools, biologists superficially freed from the UNIX shell tyranny.
- Everything still in plain text (pretty much), resources spring up like mushrooms absolutely everywhere.
- Much confusion, too many services, no way of finding them.



The time of the Grid (now)

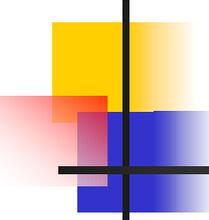
- Grids appear. No one really knows what they are. Programmers now confused as well.
- Attempts at standardisation, or possibly at using every available acronym under six letters. It's hard to tell sometimes.
- Data formats move to XML like swarms of curious angular bracketed bees.

- 
-
- Exposure of software, data and other resources as programmatically accessible service.
 - Attempts to link these services into workflows.
 - Other related attempts to automatically locate these services.
 - Very little done to allow users to interact with the resultant systems.



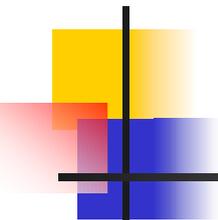
Where does ^{my}Grid fit in?

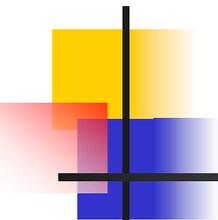
- Middleware to make a Grid system or systems usable by naïve or non expert users.
- Ontologies and reasoning.
- Service metadata and composition.
- Personalisation.
- Interface presentation.



Ontologies and reasoning

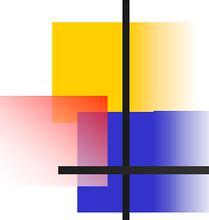
- Handle complex questions from users, working in the users' natural conceptual form.
- Used especially to locate and filter resources and data, thinning down the available choices before passing the final decision to the end user.
- I.e. tool or data set discovery.

- 
-
- Work primarily from Carole Goble's group in Manchester.
 - Based on DAML+OIL currently.
 - Focusing particularly on descriptions of bioinformatics services to drive selection and location of such services.
 - <http://www.mygrid.org.uk/myGrid/web/components/Ontologies/>



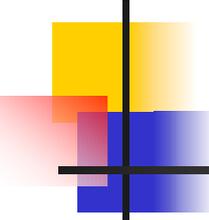
Service metadata and composition.

- Grids expose applications and data as services.
- myGrid components can assign metadata to these services, annotating them.
- This annotation can then drive service composition in workflows, the grid equivalent of hacking a PERL script to chain command line tools together.
- Annotation can also be used to discover services in the first place.



Workflows

- Intended to capture a high level scientific process.
- Originally used WSFL (Web Services Flow Language)...
- ... but now uses Scufi (Simple Conceptual Flow Language)
- Support for editing and enacting workflows.



Scufl

- Scufl is to WSDL as Java is to assembler.
- One to one mapping between Scufl language elements and the model of a workflow in a naïve user's mind (most of the time).
- Therefore easy to deal with, edit and visualize.
- <http://www.sourceforge.net/projects/taverna>

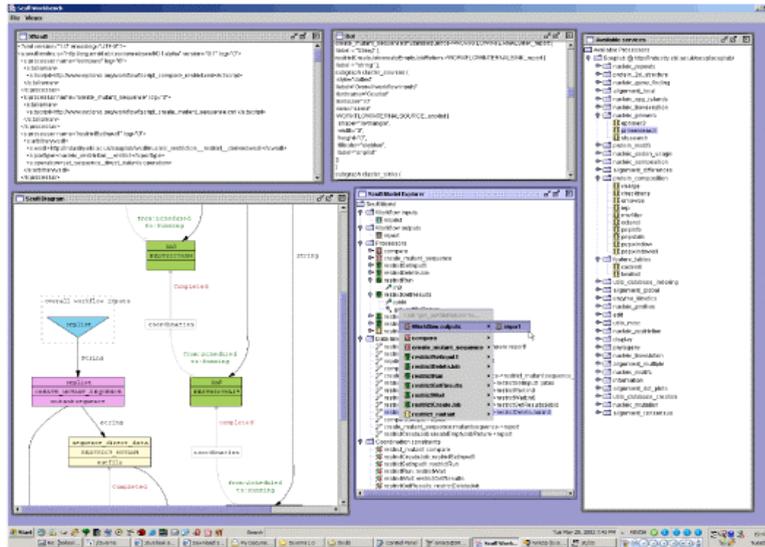
...and now, some pictures.

Early screen grab of the Scuff Workbench tool

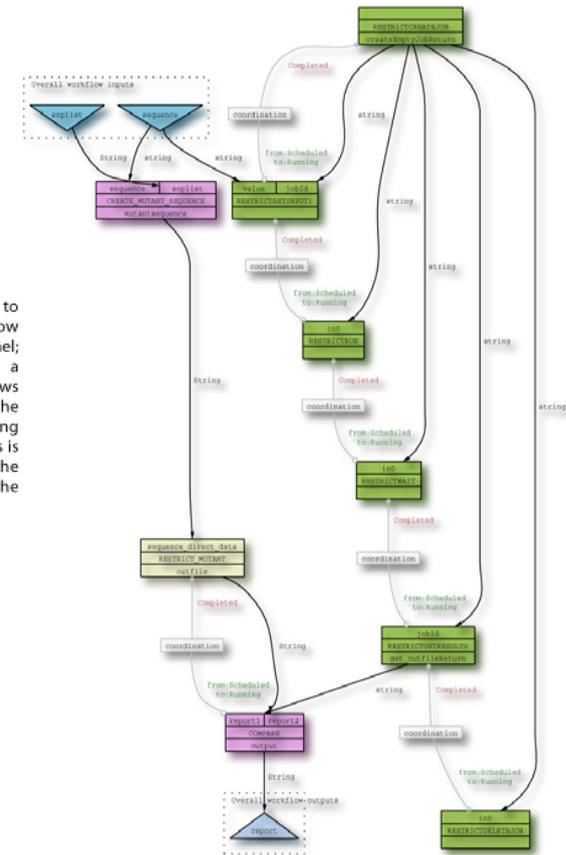
20th May, tmo@ebi.ac.uk

A textual view of the XScuff description corresponding to the current workflow; this is what the editor will save if you select the 'save as XScuff' option from the file menu.

Graph view of the current workflow; this uses the dot text from AT&T, running it locally on the client machine and making use of the results back into the component.

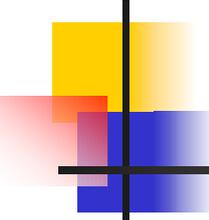


The dot text used to generate the graphical view can be shown in its own window.



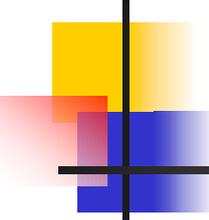
Available processors to add into the workflow are shown in this panel; right clicking on a processor node allows the user to add it to the current model. Linking of inputs and outputs is achieved through the explorer panel in the center of the screen.

Explorer tree view of the current workflow - this includes external sources and sinks, processors, data constraints and coordination controls. Removal of components can be done using the context menus on each element in the tree



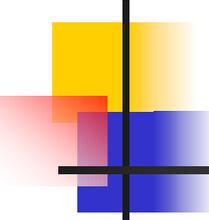
Personalisation

- Information repository stores workflow instances, results, input data, provenance etc centred around the concepts of users and experiments.
- Automatically records metadata about activities such as workflow enactment or data editing.
- Attempts to keep the eScience equivalent of a traditional biologists lab book.



Interfaces

- Workflow editing and enactment from a specialised workflow manipulation interface.
- More generic GUI based on the NetBeans environment.
- Custom web portals can be built using the Talisman RAD tool.



Availability

- Some myGrid components are readily available to download and use.
- Enactment engine, Scufi workflow language support, Talisman UI builder, Soaplab services and service construction toolkit.
- Other more research oriented components lurk deep within the CVS repository and have not yet escaped, hopefully they will soon.
- Major demo at ISMB2003, so a concurrent full release is likely.