Usage of the Astro Runtime

Noel Winstanley
nw@jb.man.ac.uk
AstroGrid, Jodrell Bank, UK
AstroGrid Workbench

- A Rich GUI Client for the VO
- http://www.astrogrid.org/desktop
Workbench - User's Perspective

- Workbench – GUI Application
  - Java WebStart / Installer / osx DMG
  - Rich user applications
    - Data discovery – Astroscope & Helioscope
    - Task Launcher – run a single task (e.g. SIAP query, remote applications, ADQL queries)
    - Workflow Builder – combine CEA applications
    - Lookout – Task Monitor and Results Access
    - Myspace browser
    - Registry Browser
  - PLASTIC – exchange data with Topcat, Aladin, Vospec, Visivo, Gaia, Specview, VOSpec,
    - Workbench does no analysis or visualization itself.
  - Scripting access to VO services (Python, Perl..)
Astroscope

- Searches all registered Cone, SIAP and SSAP services.
- Displays a concept-graph of results.
- Organized by Service; offset from search position.
- PLASTIC integrated.
- Demonstration
Registry Browser

- Google-style keyword querying
- AND, OR, NOT
- Caches Results.
- Hyperlinks to related resources and webpages
- Demonstration
Remote Applications

- Common Execution Architecture – CEA
  - Described by an IVOA Note.
  - Working system, with an installed base

- Uniform method of describing, deploying, and providing access to remote applications; such as:
  - Dataset access - Querying a catalogue database or image collection
  - Data processing - X-matching, source extraction, simulation

- CEA applications can be invoked from client scripts, UI, and server-side workflows
  - Asynchronous – invoke, notify, control.
  - Staged results – to myspace, ftp server, ..

- Standardized as IVOA 'Universal Worker Service'
Task Launcher

- Uniform form-driven interface to services
  - Cone
  - SIAP
  - SSAP
  - CEA (ADQL and others)
- Includes an ADQL builder
- Shields users from details and differences between protocols
  - adds functionality lacking in particular protocols.
- **Demonstration**
- Plan to add SLAP, latest SSAP changes, Skynode.
Lookout

- Displays execution progress and results from:
  - Task Launcher tasks
  - Workflows
- Single place to look for progress information, execution transcripts, results.
- PLASTIC integrated – local and staged results can be passed to applications.
- **Demonstration**
Myspace Browser

- Manage, upload, download, relocate files
- PLASTIC – load files in selected viewer
- Demonstration
Astro Runtime – In one Slide

• A library of virtual-observatory functions and clients.
  – scope: integrate all VO standards, popular ad-hoc services, suitable helper functions

• Library is exposed as a desktop service
  – accessible from almost all programming languages
    • XMLRPC, HTTP, RMI
      – trivial to install: Java WebStart; standalone; embedded.
      – minimal setup – no compilation or native libraries

• Library design uses consistent abstractions and types
  – cleaner API, fewer special cases, shallow learning curve, procedural / object based design.
  – Insulates clients from change and detail.

• Shared component
  – single signon, single configuration.
  – cached registry queries, other data.
All variants are available webstartable (except ASR), and as installers, and as embeddable libraries.

<table>
<thead>
<tr>
<th>Variant</th>
<th>Size</th>
<th>Plastic Hub</th>
<th>Access to VO services</th>
<th>Dialogs</th>
<th>Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workbench</td>
<td>18M</td>
<td></td>
<td>AG, CDS, NVO, IVOA</td>
<td>myspace</td>
<td>AstroScope</td>
</tr>
<tr>
<td>ACR</td>
<td>16M</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASR</td>
<td>12M</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plastic Hub</td>
<td>3M</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What can it do?

- **IVOAA**
  - Registry v1.0, - query, xquery, resolve.
  - Cone, SIAP, SSAP
  - ADQL Translator
  - SkyNode (almost)

- **AstroGrid**
  - CEA & Workflow: query, build, execute, monitor
  - MySpace: read, write, list, create, delete

- **CDS** – Simbad, Vizier, coordinates, UCD.

- **PLASTIC Hub** - client-side application messaging

- **UI** - control workbench user interface, display dialogues to prompt for input
How it all fits together...
Python: XML-RPC

1. Import xmlrpc library
2. Read AR configuration file
3. Construct xmlrpc endpoint
4. Create client
5. Get reference to service
6. Call service function

Sample Code:

```python
#!/usr/bin/env python
# Noel Winstanley, Astrogrid, 2005
# minimal example of connecting to acr and calling a service.
import xmlrpclib
import sys
import os

# parse the configuration file.
prefix = file(os.path.expanduser('~/astrogrid-desktop')).next().rstrip()
endpoint = prefix + "xmlrpc"
print "Endpoint to connect to is", endpoint

# connect to the acr
acr = xmlrpclib.Server(endpoint)

# get a reference to the registry service from the acr.
registry = acr.astrogrid.registry

# call a method
print registry.getResourceInformation('ivo://org.astrogrid/Pegase')
    # returns a struct of data

print registry.getRecord('ivo://org.astrogrid/Pegase')
    # return the xml of a registry entry (string)

print registry.resolveIdentifier('ivo://uk.ac.le.star/filemanager')
```
Astro Runtime – Improvements

- Debugged and Tested
- Registry Client Improvements
  - Supports new Registry Schema (v1.0)
  - Streaming
  - Caching
  - Xquery based
- Polished up PLASTIC Hub implementation
- VOTable Manipulation (STIL)
- ADQL Query Builder Dialog
- Split workbench into family of variants
  - Hub, ASR, ACR, Workbench
  - Reduced download size
  - installer, library, OSX .app and webstart packagings
Applications using the Astro Runtime

Searches the registry
Queries SIAP services
Saves to MySpace

Launches CEA apps on HPC resources

Browses MySpace

Lots! e.g. AstroScope:
Searches registry
Queries SIAP, Cone, SSAP services

Topcat

Workbench
Scripted uses of AR

- **VO Commandline**
  - unix-ey small composable commandline programs.
  - `vols`, `voget`, `voput`, `reg-query`, `ls-jobs` ...
  - implemented as Python scripts calling the ACR

- **Python workflows (Eduardo Gonzalez)**
  - script contains control flow
  - performs work by querying DSA servers, SIAP services, and running CEA applications via AR
  - more interactive development than batch JES workflows
  - Integrate existing non-VO programs into the workflow
  - same could be done in Perl / IDL / ...
  - Quite advanced for the average astronomer.
AR - Future

• Maintain backwards compatibility
• Documentation, examples, recipes
• Grow & support the user / developer community
• Myspace Performance, Migrate to VOSpace when ready
• Add
  – missing service type – CDS, SLAP, SkyNode
  – other useful astronomy webservices – NED, etc.
  – Expose the Astroscope engine – bulk query.
• Contributions
  – VOEvent Module – Alasdair Allen.
  – Publish DALClient ?
• Track developing standards (VOStore, SSO, TAP)
• Refine & extend workbench UI (RegistryScope...)

Noel Winstanley - nw@jb.man.ac.uk
AR - Issues

- Some open questions – similar to those of PLASTIC
  - Unsurprising, as built on same technology
  - Hope to come up with consistent solutions.
- Asynchronous event notification (XMLRPC clients)
  - Assume they're PLASTIC registered
  - Define a new plastic message for callbacks
- Security & Access Control to Hub / AR
  - Decide the appropriate place on the Security Spectrum
  - Prefer to keep things simple.
    - Existing solutions – Firewalls, Java Permissioning.
Contacts and References

- Noel Winstanley - nw@jb.man.ac.uk
- John Taylor - jdt@roe.ac.uk
- Workbench - http://www.astrogrid.org/desktop
- Astro Runtime - http://www.astrogrid.org/desktop/astro-runtime