

VO Codebases and Repositories

Mark Taylor (Bristol)
Applications Working Group

IVOA Interop
ESAC

20 May 2014

`$Id: repos.tex,v 1.12 2014/05/18 15:41:41 mbt Exp $`

Session Plan

- Introduction (*input from Apps list, Euro-VO, Paul Harrison*)
 - Context and current status
 - (Some) models for collaborative code management
 - ▷ Status quo, AstroPy, Volute, SciSoft
 - Discussions to date
 - Considerations and open questions
- Open discussion
- Next steps

Context

● Stimulus

- Heidelberg Interop Time Domain focus session (May 2013): Mario Juric (LSST) disappointed not to find a one-stop shop for IVOA-blessed implementations of VO standards, specifically in Python

● Current status

- We mostly do have high-quality implementations
- They are not particularly easy to find
- They don't necessarily integrate well together
- Cross-project collaboration is not always easy
- Code may not survive the death of its host project

● Opportunities:

- **Distribution Management:** Should we be providing easier entry for VO users? (*"Where can I download the VO?"*)
- **Collaborative Development:** Should we be working better between projects?
- Can some kind of shared codebase or common repository help?

Models

Consider a few example models for centrally organised codebases:

- Status quo
- AstroPy (careful curation)
- Volute (light/no curation)
- SciSoft (distribution)

Maybe there are other instructive examples?

Example: Status Quo

What we're doing now:

- Most (but not all) code open source
- Most (but not all) code in publicly-readable repositories
- Code mostly organised at level of national project, institution, or product
- Mostly no easy mechanism for inter-project collaborative development
- Many different Version Control Systems (VCS), build systems, documentation types, ...
- Users (have to) locate and install individual items they want from different places
- No central organisation/effort required

Example: AstroPy

AstroPy: carefully curated, collaboratively developed package

What is it? Git/github repository + Coordinated releases + User website + Mailing list

Content: Python library code with comprehensive documentation

Main aim: Provide robust astronomy libraries to python users

Owned/run by: The AstroPy collaboration (2013A&A...558A..33A)

- Wide, but controlled, astro subject range (N-d & tabular data, I/O formats, units, coordinates, fitting, cosmology...) Some VO content (VOTable, SAMP, Cone Search)
- Careful curation
 - ▷ Formal processes for accepting contributions, developing new features, documentation standards, release and packaging, ...
 - ▷ Considerable effort required from team of three coordinators
 - ▷ High-quality, well-documented distribution produced
- “Affiliated packages” for code aspiring to enter AstroPy (quality requirements even for these)
- Wide participation (~60 contributing developers)
- Active mailing list, involved community
- Python a special case? Astronomer users, Python versioning issues.

<http://www.astropy.org/>

Example: Volute

Volute: informal repository to facilitate IVOA-related work

What is it? Subversion repository, hosted on Google Code

Content: Miscellaneous IVOA-related items, mostly documents and data model descriptions

Main aim: Collaboration tool

Owned/run by: IVOA (without formal endorsement)

- Set up by Norman Gray in 2007 for IVOA use
 - ▷ Set up because it seemed like a good idea, no formal TCG/Exec involvement
(this approach often seems to work quite well)
- Minimal active curation
 - ▷ Loosely organised by WG
 - ▷ No common build procedures, document standards, acceptance criteria etc
 - ▷ To make contributions, ask an Owner (e.g. Norman) for write access
- Used by many WGs to facilitate multi-author, version-tracked standards development etc
 - ▷ >30 items (documents, DMs, vocabularies, ...), 25 committers, 2600 revisions

<https://code.google.com/p/volute/>

Example: SciSoft

SciSoft: ESO-curated distribution of third-party astro software

What is it? Downloadable distribution + Website

Content: Various libraries, applications, packages, infrastructure used by astronomers (IRAF, MIDAS, ds9, Skycat, fv, PGPLOT, CFITSIO, HyperZ, some VO tools, ...)

Main aim: Provide common installed software environment for astronomers

Owned/run by: ESO

- Considerable curation effort (build+distribute) required
- Minimal or zero effort/participation by code developers
- *Possibly moribund — current release is March 2012, Fedora 11 only*

<https://www.eso.org/sci/software/scisoft/>

Pros and Cons

Depending on the details, adoption of a common codebase may have:

- Potential advantages
 - ▷ Easier for third parties to find/use/integrate VO software
 - ▷ Encourages/facilitates contributions from non-core developers
 - ▷ Encourages communication between developers on different projects: sharing code, libraries, approaches, ideas, best practice
 - ▷ Community ownership may result in better support
 - ▷ Community can take over “orphaned” code when parent project/funding ends
 - ▷ Easier to start a new software item
- Potential disadvantages
 - ▷ Reduced control of code by “owner”
 - ▷ Less obvious ownership/credit for main developers
 - ▷ More heavyweight release process, less control over release schedule
 - ▷ May require use of non-favourite VCS/build system/doc format/...
 - ▷ Integration effort required to transfer in existing code
 - ▷ Sticky Lump

Discussions To Date

Apps mailing list (post + responses Feb 2014)

- We should be doing it
- VCS choice discussions
- Volute has been successful
- What goes in: client? server? libs? not python?
- No actual volunteers to contribute specific code

CoSADIE Tech Forum 3 (Trieste March 2014, led by Paul Harrison)

- General enthusiasm
 - ▷ some effort volunteered
 - ▷ some code tentatively offered
- VCS choice discussions
- Suggestion to just go off and do it, see who joins in

Other informal discussions?

Open Questions 1

- Do we want to set something up at all?
 - ▷ A repository?
 - ▷ Multiple repositories? By language? By function (e.g. client/server, library/app)?
 - ▷ Or just encourage use of existing technology (Volute? Github? Maven?)
- What do we want to achieve:
 - ▷ Cooperative development? Managed releases? Sharing code?
- Software curation practices
 - ▷ Are there controls on what's allowed in? by scope? by quality? by licence?
 - ▷ Is code with duplicate functionality allowed/encouraged/discouraged?
 - ▷ Is documentation required to be in a particular form?
 - ▷ Coding standards? Unit/system tests?
 - ▷ Must all code be compatible (e.g. use compatible library versions)?
- Version control system and hosting service (* *Warning: religious issue* *)
 - ▷ git/hg/svn? github/bitbucket/google code/self-hosted?
- Build/release mechanics
 - ▷ Require single build system?: maven/ant/make/...?
 - ▷ Single synchronized release for all contents? For all libraries?
Or leave build/release to individual product “owners”

Open Questions 2

- Interaction with other repositories
 - ▷ Aim towards AstroPy integration for some/all VO python code?
- Status and organising principles
 - ▷ Is the repository officially owned by the IVOA? Is contained code thereby endorsed?
 - ▷ Who's in charge: nobody? benevolent dictator? committee? TCG? a WG? whoever's prepared to do the work?
- Decide policy up front or just set it up and see what happens?

Participation

- Which projects/code bases are willing
 - to use a central repository for code development?
 - to contribute curation effort?
- How much participation is necessary to make it worthwhile?
 - Just enough to get it started? (*“Build it and they will come”*)
 - Enough to constitute a worthwhile download for users?

Discuss!