How to use the VO from IDL

An Example on how to Combine Existing Software with the VO

Iliya Nickelt, AIP

German Astrophysical Virtual Observatory (GAVO)

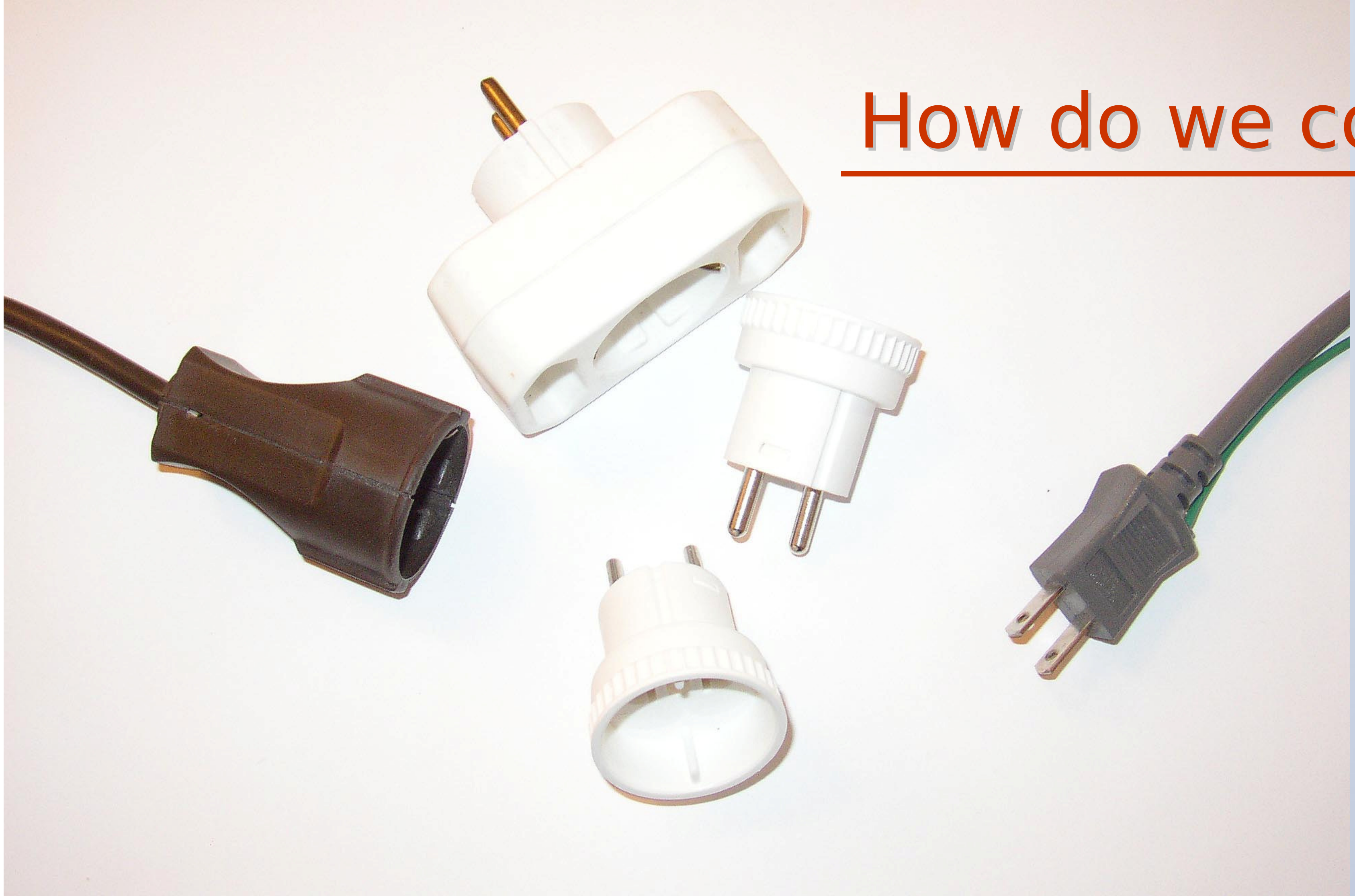
Overview

- Introduction: Principles of combining existing software with the VO
- IDL, Java Bridge
- IDL Aladin Connection
- IDL VOlib: The Basic Routines

A Fundamental Problem

- New technologies have many advantages – such as the VO.
- But existing users are experienced with their very own software.
- Specialised applications, widely used and sometimes very advanced.
- How can you keep the old and make use of the new advantages?

An Old Network Question



How do we connect?

The Four Paths to Happiness

To combine old and new, you can:

1. Write a *native language library*, i.e. use existing routines and libraries to extend your software package / language of choice by the necessary VO routines;
2. Include existing binary *libraries* (e.g. from JAVA);

four options - continued

3. Connect to existing binary *tools* (via an object bridge, Plastic, ...);
4. Use *command line* interface tools via system calls, share information via stdout and files.
5. ... or combinations of 1 to 4

Obstacles

- Binaries may be system dependant
- Binary inclusion is non-trivial for users: Missing dependencies (!), OS paths, settings.
- Development is rapid, so external tools may change (... → dependency troubles)
- It may be difficult to identify and choose among existing solutions (no accepted central VO code repository)

Native libraries

- Native libraries are best
- ...but may need considerable time to implement.
- Skills are also desirable (robustness, speed).
- In a pre-compiled environment, an extension of existing code may be very difficult (source code needed, legacy software, ...)

What is IDL?

- „Interactive data language“
- FORTRAN history, but also object orientated and tool elements.
- Good for quick software development (interpreter functionality) and for handling and displaying large arrays.
- Popular in astrophysics, usage is still growing (it seems). Used for reduction and analysis.

IDL cons

- Commercial and (too) expensive.
- License system makes distributing the code difficult (but: virtual machine), also bad for automated processes (grid computing).
- GDL might work sometimes (but maybe not in this example).
- Company has in recent years been slow to implement network libraries, XML etc.

IDL Java Bridge

- Available for IDL V6.0++, for almost all OSes
- Uses the Java Virtual Machine to launch Java programs (“Java reflection API”)
- JVM, options and classpath are specified in a setup file (→ user intervention needed)
- JAVA methods of a defined class become methods of an IDL object, variables are translated
- Certain restrictions apply (variable types, method name translation,…)

Alternatives

IDL has other connection methods:

- “call_external”, ActiveX.
- extensive libraries exist for file exchange, e.g. FITS.
- OS independent system calls (“spawn”).

Aladin IDL interface

Written by Thomas Boch, Laurent Cambrésy, Bernd Vollmer (CDS).

Uses the methods of the Aladin class to:

- Load into and retrieve from Aladin planes, images and tables;
- Execute Aladin script commands;
- Additional Aladin specific commands (ROI, colortable, plane list, ...).
- OS independent (to be released soon)

- Demonstration -

In this demo, we will:

- Launch Aladin
- Locate and download some data from the VO.
- Transfer the data (image planes) to IDL.
- Work with it in IDL.

VOlib (V0.2b) overview

Written by Christopher J. Miller (NOAO/CTIO)

- Uses multiple JAVA classes and corresponding IDL wrappers.
- VO Table reader (“readvot.pro“), SIAP (“siapcall.pro”)
- VO Registry search (“call_registry.pro“), cone search (“conecall.pro“)
- Additional: NVO Open Sky Query, name resolver

VOlib status

- Not OS independent (yet...)
- Needs a lot of libraries
- beta stage in some parts
- Additional features desireable: Spectra, other protocols (SAMP, ADQL), ...
- Routines for *publishing to the VO*: Write VOtables, database connection, VO-compatibility check of metadata

IDL VOlib examples

```
IDL> call_registry, str=reg, '2MASS',/SIAP
IDL> print, reg.url
  http://irsa.ipac.caltech.edu/cgi-
  bin/2MASS/IM/nph-im_sia?type=ql&ds=asky
IDL> siapcall, str=img_2m, 217.5,35.0, 0.16, $
  url=reg.url, root='2mass'
```

VOlib examples II

```
IDL> call_registry, 'parallax', /cone,
  str=cone
IDL> print, cone.title, format = '(a80)'
  Hipparcos Space Astrometry Mission
  All-Sky Compiled Catalogue of 2.5 Million
  Stars
  Bright Star Catalog
  Gliese Catalog of Nearby Stars (3Ed)

IDL> print, cone[2].url
  http://heasarc.gsfc.nasa.gov/cgi-
  bin/vo/cone/coneGet.pl?table=bsc5p
IDL> conecall, 180,1,10, str=str, $
  url=cone[2].url
```

Summary

- There is no simple solution when combining existing packages with the VO
- The fastest way is to use JAVA libraries for powerful methods and access to tools
- VO Development for IDL has started and can be used, but it is still at an early stage.

Authors & Sources

volib (V0.2b):

Christopher J. Miller (NOAO/CTIO)

http://www.ctio.noao.edu/~chrism/VOlib

Aladin/IDL:

Thomas Boch, Laurent Cambrésy, Bernd Vollmer (CDS).

http://eurovotech.org/twiki/bin/view/VOTech/AladinIDL

VOcli:

Mike Fitzpatrick, NOAO (NVO)

http://iraf-nvo.noao.edu/vo-cli/index.html

GAVO

- German VO, sponsored by the BMBF
- University of Heidelberg (ZAH/ARI), MPE, AIP, University of Tübingen, Technical University München
- Surveys: ROSAT, RAVE
- Theory: Millenium, model spectra
- Smaller Data Archives, Grid computing, ...
- Collaborates with the Bulgarian VO (Wide field images and data base)