



# Registry Applications Data Model

Paul Harrison ESO

Part of the “tiger team” effort

# Goals

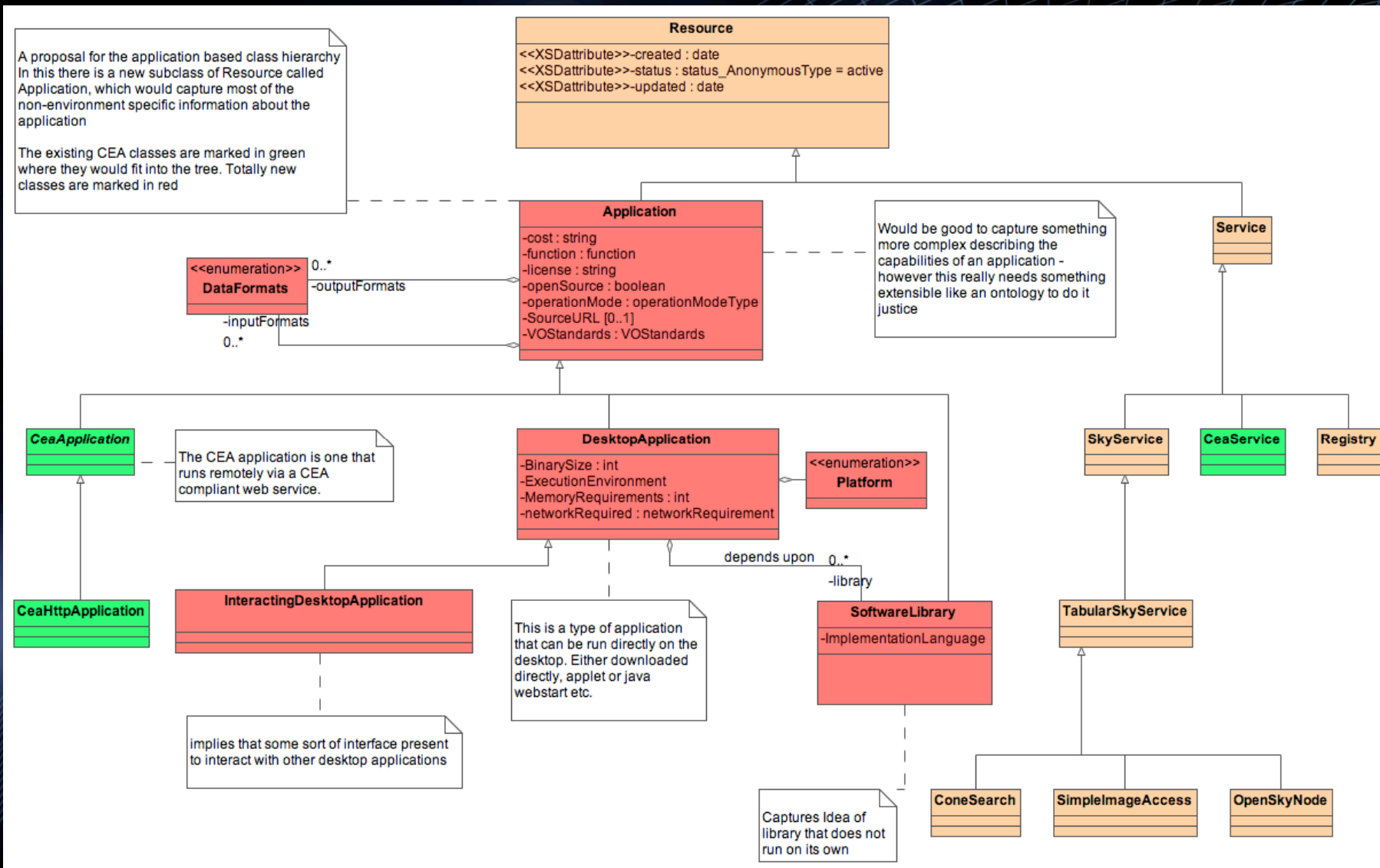
- Support general application specific metadata - e.g. source code language
- Have information on what VO Standards and data formats an application can support
- Support Concepts of
  - Desktop application
  - Software Library
  - Applications that can be called as services.

# General Metadata

- These were enumerated at the last interop which generated [this document](#)
- Some highlights
  - Function - what does the application do?
  - Dependencies - what else is necessary?
  - VO Standards - what
  - Data Formats
  - Platforms



# UML Model



# Overview of rest of talk

- This talk will concentrate on the part of the UML diagram coloured red
  - Represented in VOApplication.xsd file.
    - Application
    - DesktopApplication
    - SoftwareLibrary
- Next session will discuss the CEA application model for remote applications.

# Schema

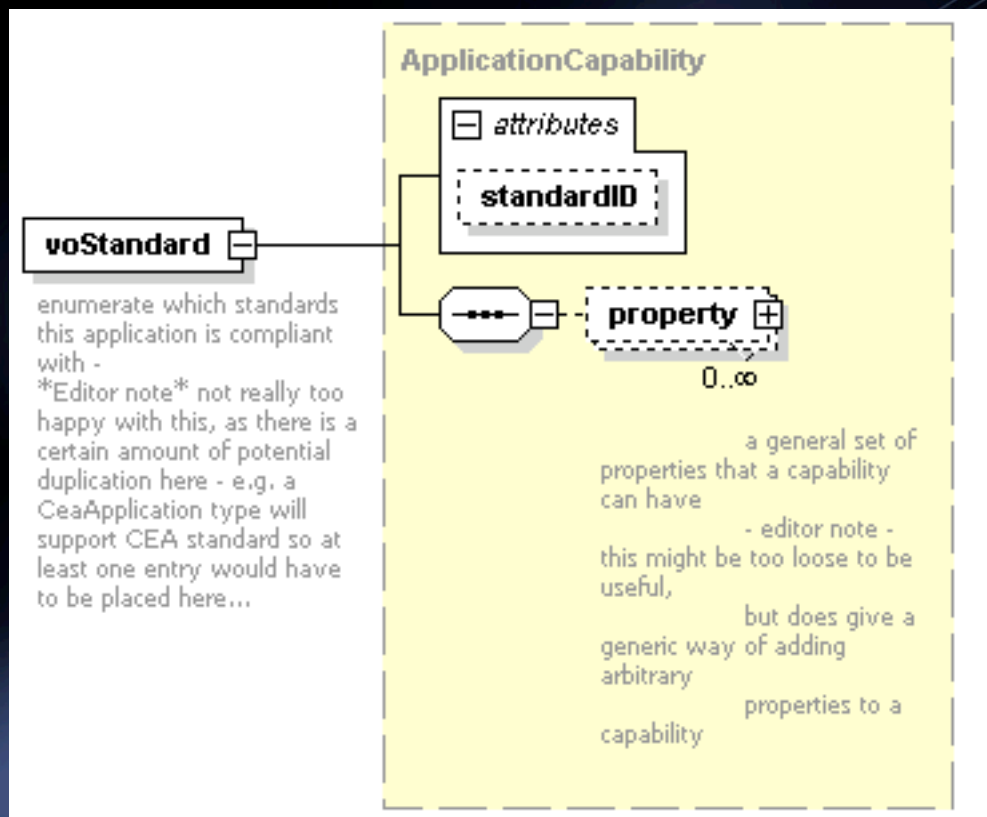
- Representing the model as XML schema is not always straightforward
- Derivation vs. aggregation of properties
- Schema rigidity vs. completeness
  - Do enumerations cover all cases?
  - Will enumerations change with time forcing new schema to be issued.
- Some concepts - e.g. what does the application do, probably better suited to Ontology approach.
- Current proposed schema attached to <http://www.ivoa.net/twiki/bin/view/IVOA/RegDMApplications>



# Programming Language

- Example of enumeration dilemma
- Currently set to C, C++, C#, FORTRAN, Java, Perl, Python, other
- Obviously not complete
- Does not contain versions
- Is it enough to capture majority of what an astronomer is interested in?
- But if we were to try to make a full model, who would curate it? - outside VO scope.

# VOStandard



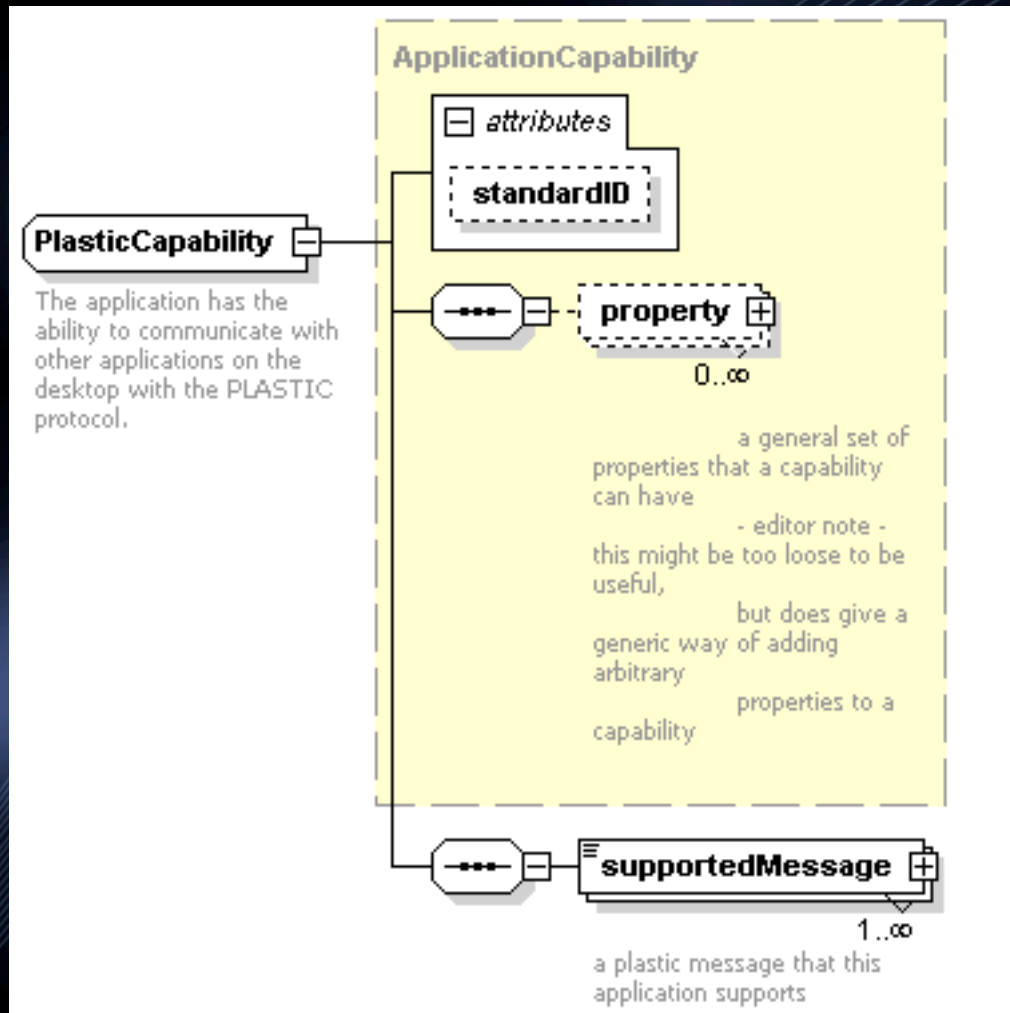
- Similar to the Service Capability
- No interface
- Generic property pair?



# ExecutionEnvironment

- XMLSpy Doc
  - Platform - The basic platform name - platform in this case is related to how to execute application - enumerated
  - Subtype(opt) free string
  - Minversion(opt), Maxversion(opt) - problematic how do you express Windows XP > Windows 95
  - Download(many) - URLs where the executable may be obtained
  - Path(optional) - in case the downloaded file is some sort of archive. The path is relative to the base of the archive

# PlasticCapability



- An example extension capability for **PLASTIC**
- Enumerates the messages that an application can deal with
- Could be used by PLASTIC hub to automatically launch required application

# Examples

- Schema design being informed by set of examples
  - Aladin - a DesktopApplication
  - Astrogrid Workbench - Another desktop application
  - STILTS - a software library
  - Sextractor - Desktop application - not Java!