university of groningen

Astro-WISE: Tracing and Using Lineage for Scientific Data Processing

Johnson Mwebaze, Danny Boxhoorn and Edwin Valentijn

Kapteyn Astronomical Institute
University of Groningen
The Netherlands

Outline

- Introduction
- Motivation
- Astro-WISE
- Framework
- Data Lineage and Data Processing
- Tracing Lineage in Astro-WISE
- Experiments [Case Studies]
- Related Work
- Conclusions and prospects

Introduction

- Data growing exponentially
  - caused by successive generations of inexpensive sensors + exponentially faster computing
  - Old and New Projects in the pipeline; Creating accelerated acquisition of large volumes of observational data
- Changes the nature of scientific computing (caused by Technology, Sociology, Economics)
  - The Virtual Observatory, a collection of data centers each with unique collections of astronomical data.
  - how to believe, share and validate data as published.
  - how to selectively retrieve and process data from multiple archives interactively
- how to allow flexibility in e-science systems to enable researchers to work with and change methods on the fly while capturing all processing activities/events
- Scientific computing is revolving around data [Jim Gray] - Move analysis to the data.

Why Lineage/Provenance

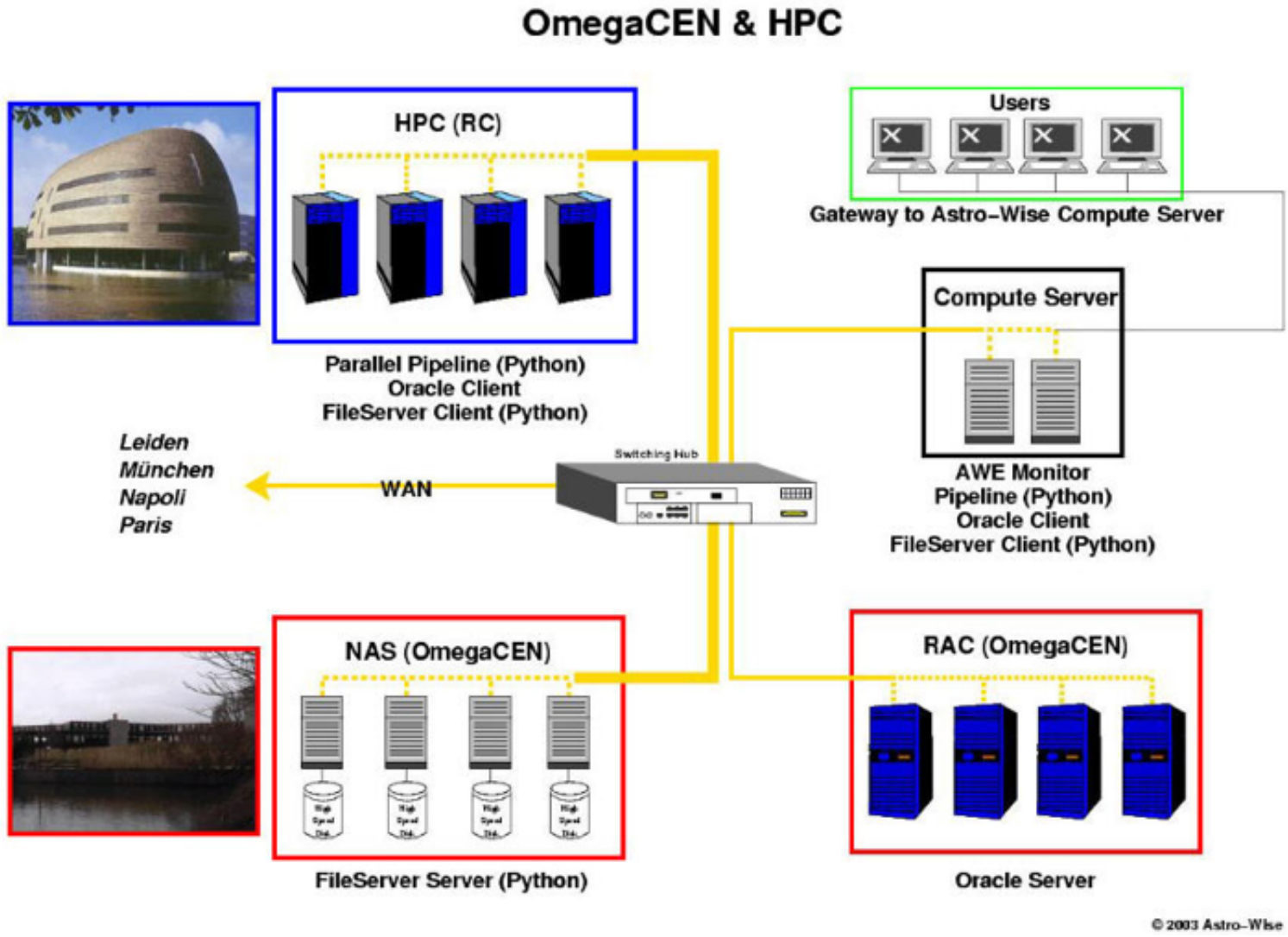
- Results validation
- Result debugging
- Reproducibility
- Repeatability
- Explanation(derivaHons,traces,prooftrees)
- Runtime monitoring (Proling,benchmarking)
- Performance Optimization(smart rerun)
- Faulttolerance,crashrecovery
- System re-design

What is Astro-WISE

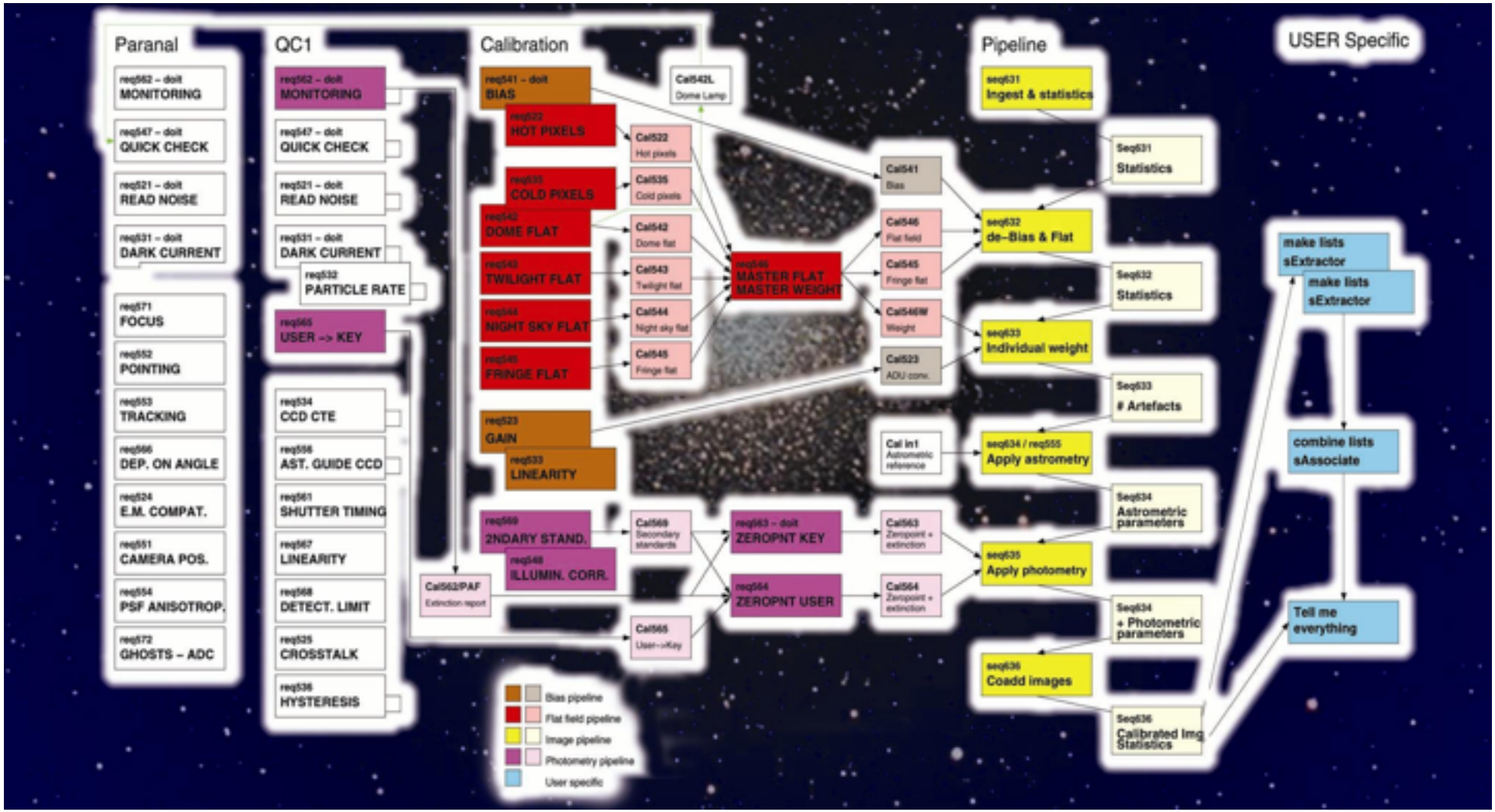
- Goal: Connect raw data to end user in VO environment
- an e-science infrastructure with fully distributed resources, which allows teams distributed over Europe
- handles optical, IR and radio.
  - Current: WFI@2.2m, INT, Subaru, ACS@ST
  - Near future: Panstarrs, VST, VISTA, Lofar
  - Far future: EUCLID
- Services
  - pipeline data reduction, calibration and re-calibration
  - Image comparisons and combinations
  - Working with source lists
  - Visualization of astronomical data
  - publishing results to the Virtual Observatory Archives

Motivation

- Wide-field imaging = vast amounts of data
  - VST sees equivalent of Southern sky with 0.2 pixels in 3 years.
  - A project like KIDS (1000 Sq Deg) has ¿10^6 8Mpix raw data images
- Handling of the data is non-trivial
  - Pipeline data reduction
  - Calibration with very limited resources
  - Things change in time:
    - Physical changes (atmosphere, various gains)
    - Code (new methods, bugs)
    - Human insight in changes
  - Working with source lists


OmegaCEN & HPC
HPC (RC)
Parallel Pipeline (Python)
Oracle Client
FileServer Client (Python)
Users
Gateway to Astro-Wise Compute Server
Compute Server
AWE Monitor
Pipeline (Python)
Oracle Client
FileServer Client (Python)
Leiden
München
Napoli
Paris
WAN
Switching Hub
NAS (OmegaCEN)
FileServer Server (Python)
RAC (OmegaCEN)
Oracle Server
© 2003 Astro-Wise

AstroWise Pipeline



the Framework

- The core of the system exploits three properties in a database environment.
  - Firstly, we apply the principle of inheritance using Object Oriented Programming (Python), where all Astro-WISE objects inherit key properties for database access, such as persistency of attributes.
  - Secondly, the linking (associations or references) between instances of objects in the database is completely maintained.
  - Thirdly, continuous growth of the database through the addition of new information or improvements made to existing information.
    - archiving or regenerate dynamically
    - exchanging methods, scripts and configuration

DataBase Design

- Persistence -
  - persistence objects is the core of Astro-WISE pipeline processing and data lineage
  - Processing is done through invocation of methods on these objects.
  - All the I/O of the pipeline processes makes use of a database
  - Data can therefore only be manipulated through interaction with the database
- Extendable Schema
  - allows extending and modifying object attributes and method definitions through inheritance and polymorphism
- Version Control (Object Versioning) and Preservation Management
  - the connection between these classes and the created objects
  - addresses object inconsistencies (mismatches) created as a result of schema modification

Tracing Lineage in Astro-WISE

- For each persistent class, a number of methods are defined which interact with the federated database. All class instantiations are automatically made persistent in the database forming an archive of all targets. To achieve this, the following major classes are implemented.
  - `DBObject` is the root class of the hierarchy of the persistent classes. This class defines the primary key `object_id` of all objects.
  - `DBObjectMeta` is the metaclass of `DBObject`. This is the class that is responsible for class creation and object instantiation. (class factory)
  - `DBProperties` is the module that defines all persistent attribute types (data dictionary) which are defined by `persistent`.
  - `DBSelect` implements a query language that is a natural extension to Python and that incorporates data lineage in the query syntax.
  -
  - $sourcelist.coadd_{f}rame.regr_{f}rame.reduced.raw.filename$
  - `DBProxy` is an abstract interface to database vendor specific operations.

Lineage capture and Storage

- lineage is captured as dependencies between persistent objects that refer to another persistent objects.
- The simplest persistent class has only one persistent attribute, like

```
class ClassA(DBObject):
  attribute = persistent('Description',
            attribute_type, default_value)
```

- `attribute` will have the type `attribute_type`.
- Atomic attribute types, such as `integer`, `string` or `float`, are translated into their database equivalents when an object is made persistent.
- If `attribute_type` is a subclass of a DBObject, the attribute will be a link to an object of that subclass.
- If `default_value` is an empty list, then the attribute will be an array of objects of `attribute_type`.

Tracing Lineage in Astro-WISE

- another Example

```
class ClassB(Object):
  e = persistent('A link', ClassA, None)
  f = persistent('List of links', ClassA, [])
  g = persistent('Link to object of ClassB')
```

- `ClassB` defines three persistent attributes:
- `e` is a link to an instance of `ClassA`,
- `f` is a array of links to instances of `ClassA` (default empty), and
- `g` is a link to another instance of `ClassB`.

Storage

- To support the storage of files in a similar way, the `DataObject` class is used.

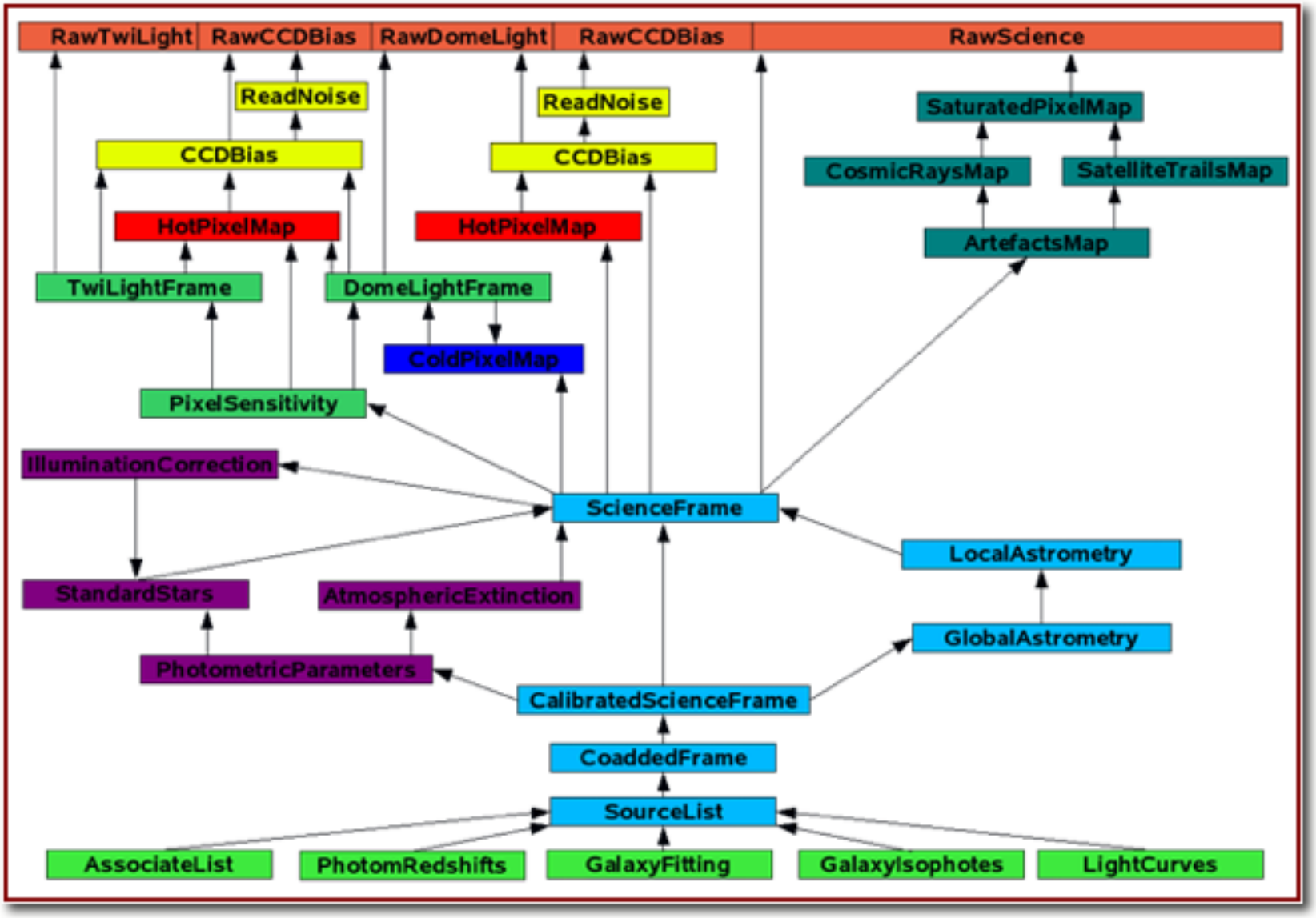
```
class DataObject(DBObject):
  filename = persistent('File part', str, '')
```

- The `filename` attribute is used to implement `store()` and `retrieve()` methods to transfer files to and from dataservers.
- Because the filename is kept in the database the storage and retrieval of files is transparent to the application.

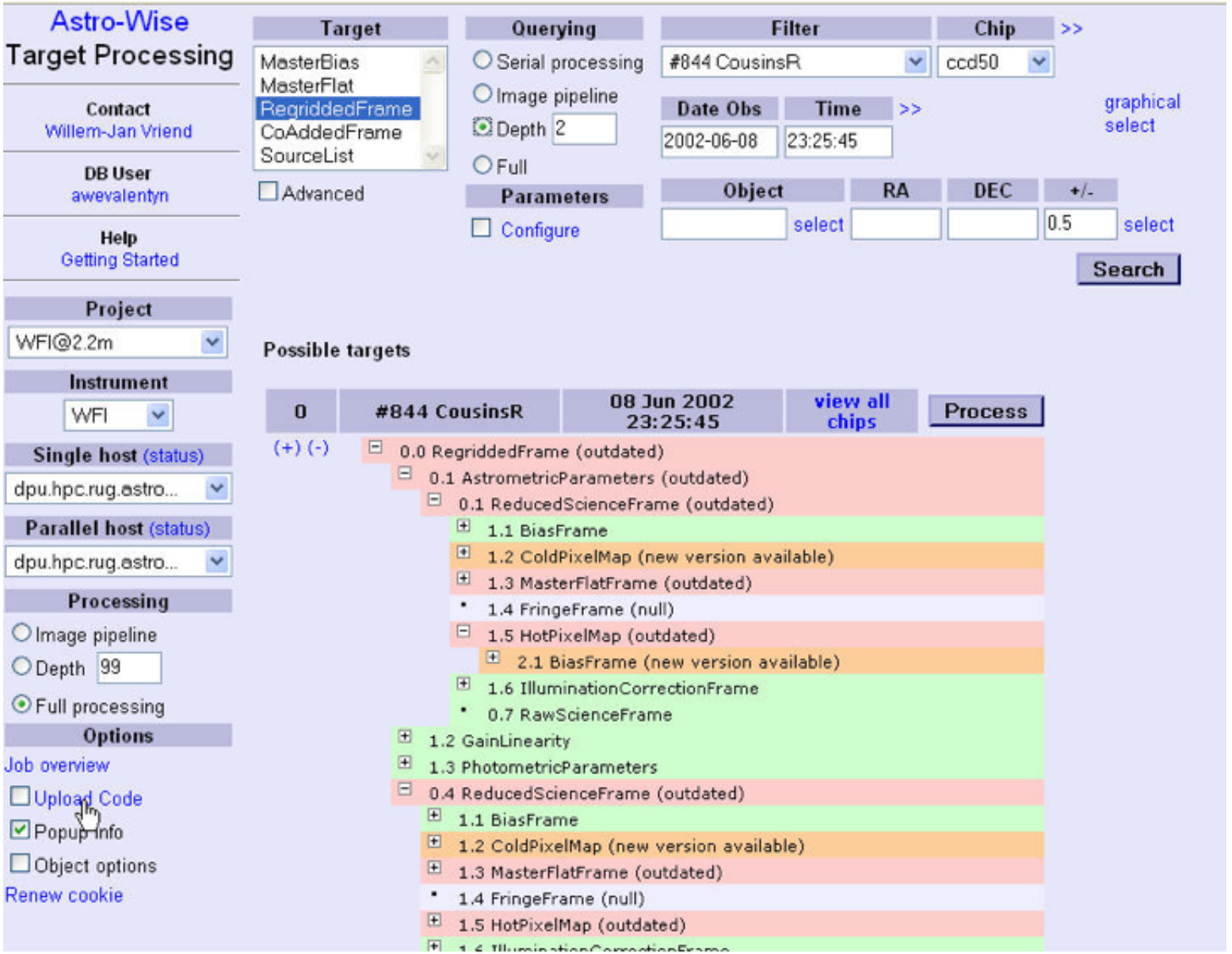
Target Processor

- backward-chaining approach while processing data
- The make metaphor -specify the result instead of the input
  - This allows the end user to trace the data product, following all its dependencies up to the raw observational data and, if necessary, to re-derive the result with better data, and/or improved methods
- Uses target Processing (TP)
  - The $Target$ processor employs the dependency logic that is constructed using lineage data
  - Targets are rebuilt in a recursive cascade only if the dependencies have changed (i.e workflow reduction).

Simplified ObjectModel



WebServcesObjectMaker



Object Viewer - info() method

```
+-Name of SourceList : SL-0000010011
+-COMBINE_METHOD: -1
+-OBJECT: field07B
+-SLID: 10011
+-associatelist: -1
+-astrom_params: None
+-chip: None
+-creation_date: 2006-03-20 13:57:25
+-detection_frame: None
+-filename: ldac.cat
+-filter: <astro.main.Filter.Filter object at 0x8df22d0>
+-filters:
+-frame: <astro.main.RegriddedFrame.CoaddedRegriddedFrame object at 0x8dd1b10>
+-globalname:
+-instrument: <astro.main.Instrument.Instrument object at 0x8df2050>
+-name: SL-0000010011
+-number_of_sources: 2236
+-object_id: '0F6E9292574FC6EEE0407D81C5061B26'
+-process_params: <astro.main.SourceList.SourceListParameters object at 0x8df20d0>
+-sexconf: <astro.main.Config.SextractorConfig object at 0x8df2150>
+-sexparam: <class 'common.database.typed_list.typed_list'>(<type 'str'>, ['DELTA_J2000', 'ALPHA_J2000'])
+-sources: {}}
+ - [some output omited]
```

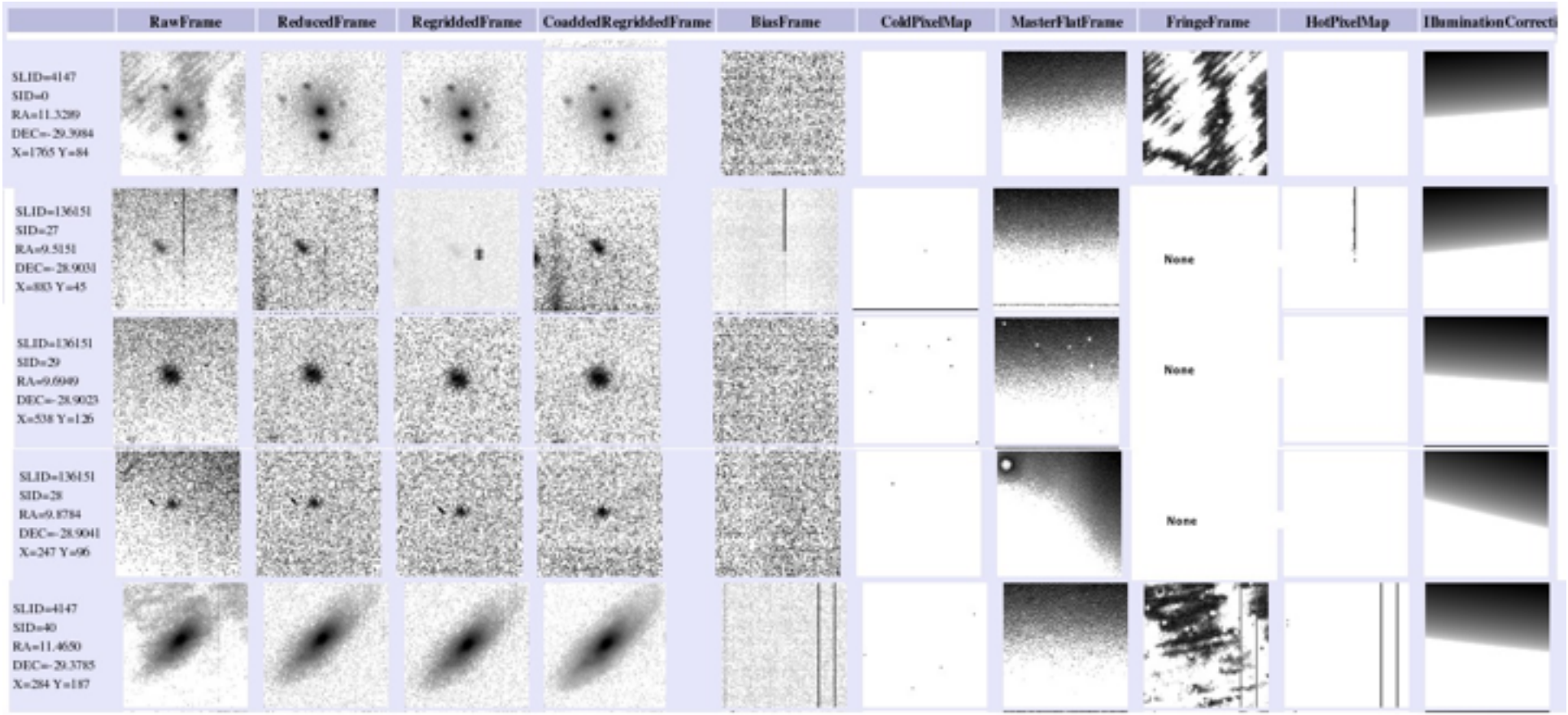
- Queries e.g sourcelist.frame.info()

Object-Viewer

```
Main-Object : SourceList    qualityview
 |+ OBJECT                | STD
 |- name                  | GAS-Sci-WVRIEND-WFI-------#842-ccd50-Red---Sci-54964.6029650-8d91ef050a6e71840937ba4a40f538
 |+ number_of_sources     | 36
 |+ sexparam              | (, ['FLUX_RADIUS', 'FWHM_IMAGE', 'MAG_ISO'])
 | +astrom_params           [level : 2] AstrometricParameters  --> QC-WVRIEND-WFI-------#842-ccd50-AstromParams---Residu
 | | +associateconf           [level : 3] AssociateConfig
 | | +astromconf              [level : 3] AstromConfig
 | | +process_params          [level : 3] AstrometricParametersParameters
 | | +reduced                 [level : 3] ReducedScienceFrame    qualityview
 | | |- filename              | (image)Sci-WVRIEND-WFI-------#842-ccd50-Red---Sci-54964.6029650.....fits
 | | | +astrom                  [level : 4] Astrom
 | | | +bias                    [level : 4] BiasFrame
 | | | +chip                    [level : 4] Chip
 | | | +cold                    [level : 4] ColdPixelMap
 | | | +filter                  [level : 4] Filter
 | | | +flat                    [level : 4] MasterFlatFrame
 | | | +hot                     [level : 4] HotPixelMap
 | | | +illumination            [level : 4] IlluminationCorrectionFrame
 | | | +process_params          [level : 4] ReducedScienceFrameParameters
 | | | +raw                     [level : 4] RawScienceFrame    qualityview
 | | | +template                [level : 4] Template
 | | | +weight                  [level : 4] WeightFrame
 | | +sexconf                 [level : 3] SextractorConfig
 | | |- WEIGHT_IMAGE          | (image) Cal-WVRIEND-WFI-------#842-ccd50-Red---Wei-54964.6031358-....fits
 | |- filename                | (image)  Sci-WVRIEND-WFI-------#842-ccd50-Red---Sci-54964.6029650-....fits
```

SubImage CutOuts

- we can trace to a pixel level
- this can be used to find extremely rare astronomical events, such as a fast moving star, near-earth solar system objects, or variable objects like ultra-compact binary systems or distant supernovae



Questions

?