# Integration of the MUSE Software Pipeline with the Astro-WISE System

We discuss the current state of integrating the MUSE software pipeline (Weilbacher et.al. 2006) into the AstroWISE system. MUSE is a future integral-field spectrograph for the VLT, consisting of 24 integral field units. The MUSE data reduction pipeline is built using the Common Pipeline Library (CPL) provided by ESO. The AstroWISE technology (Valentijn et. al. 2007) integrates data lineage, data persistence, distributed processing, and large file storage into a information system. To integrate the MUSE pipeline, its metadata is used to build persistent objects for storage in the AstroWISE system. It is thought that this method could provide a convenient and quick method to implement future pipelines into AstroWISE. Current work on the integration includes handling multiple IFUs, completing the pipeline integration, and use-case development.

James Pizagno (1), Ole Streicher (2), Willem-Jan Vriend (1)
1 Kapteyn Astronomical Institute, University of Groningen, PO BOX 9700 AV, The Netherlands
2 Leibniz-Insitut für Astrophysik Potsdam, An Der Sternwarte 16, D-14482, Potsdam, Germany

## Astro-WISE

The Astro-WISE system has been developed to handle the large collection of data produced by modern Astronomical experiments. It integrates data modeling, large data storage, distributed data processing, and data lineage into a single system. The system is designed to allow users to mine for knowledge. Using an integration of software and hardware, scientists can retrieve and process data in a single environment. All the steps of the data processing are linked, allowing in-depth analysis questions to be answered. The system was initially developed to handle data from OmegaCAM, but has currently been extended to other astronomy and non-astronomical projects.

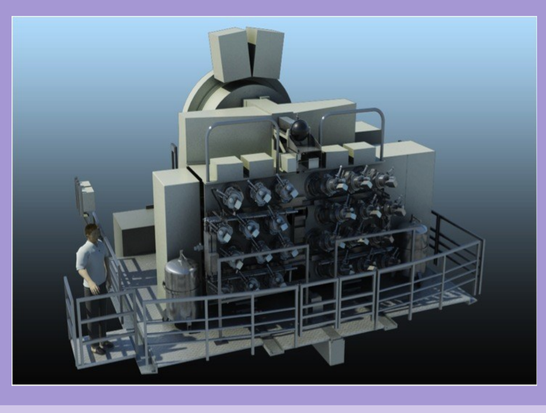
The Astro-WISE system includes 1.6 PB of data storage, 10 Tflops of computing power, an Oracle RDBMS, and a variety of built-in analysis tools. A DBViewer allows a web-based view of processed data. The system has the functionality to publish data to colleagues and handle access within a given project. The implementation of AstroWISE in Python allows the system to be easily distributed to partners, and integrated into existing data reduction pipelines.

Web link: http://www.astro-wise.org/

## MUSE

MUSE is an Integral Field Unit spectrograph consisting of 24 IFUs that will obtain a full spectrum for each pixel in the field of view. It will start operation on the ESO-VLT in 2013. Each data cube size will be 1570 MB. The MUSE software pipeline currently uses the ESO-CPL library. The MUSE project presents a large data processing, data storage, and data distribution problem. Since AstroWISE has a system that has already been developed to handle large data storage and large data processing, we detail the efforts so far to integrate the MUSE software into the AstroWISE environment.

Web link: http://muse.univ-lyon1.fr/



## Integration

MUSE has an interface to its CPL-based pipeline using Python-CPL (see ADASS poster on Python-CPL (P148)). The MUSE pipeline has XML files defining data objects and associated attributes, which are used as inputs for a given MUSE recipe. Each Astro-WISE class is built from the same XML object definitions, which has an associated processing parameters which are persistent objects and retrievable. Each object has associated methods for data storage, data processing using the MUSE recipes, and object persistence. The current implementation produces all necessary data outputs given a single MUSE pipeline step, where each data model has an associated methods class and processing parameters class, and inherits methods for persistence and data storage. A task is defined for each step in the pipeline, which also handles multiple outputs for a given MUSE recipe.

MUSE XML file:

```
<plugin name="muse_bias" type="recipe">
   ....
   <frame tag="MASTER_BIAS" group="product"   mode="master">
   ....
   <parameters>
   <parameter name="nifu" type="int">
      <description>IFU number to handle</description>
      <default>99</default>
   </parameter>
   ....
</plugin>
```

MUSE Astro-WISE:
Parse XML into Astro-WISE Python classes

Resulting MUSE-AstroWISE Classes:

```
class   MASTER_BIAS(DataObject ,MASTER_BIAS_methods):
   process_params = persistent("Processing parameters recipe_bias",recipe_bias,None)

class  recipe_bias(DBObject):
   ....
   nifu = persistent("IFU number to handle",int, 99)
   ....
```

An example step in the MUSE pipeline is the generation of a master bias from raw input bias frames. The master bias is handled through a AstroWISE task which runs the MUSE recipe 'muse_bias' and generates an output MASTER_BIAS, which includes a unique FITS image and processing parameters. Processing parameters, such as IFU number and sigma-clipping limits, are automatically stored in a RDBMS. These parameters can be used to select and retrieve specific result frames. The output FITS data files are stored on a MUSE-AstroWISE data server , which allows the data to be accessed by collaborators in different locations. The methods associated with a given MASTER_BIAS include checking preconditions, linking of MUSE recipe parameters with persistent objects, MUSE recipe running, and unique data naming.

## Conclusion & Future Work

The integration of MUSE recipes into Astro-WISE is an on-going process. The Astro-WISE tasks currently run the MUSE recipes, processing parameters are persistent, and data in stored in a MUSE data server. The units of tasks have been developed in such a way to allow distributed processing of MUSE pipeline steps for a given IFU unit data reduction.

Future work will include combining results from different IFUs, quality assurance parameters, full automation, and deployment to the MUSE team. The current pipeline runs on an individual IFU basis, but later steps must include combining multiple IFUs into a single data unit. Quality assurance parameters for each analysis step will be made persistent. Full automation from MUSE XML files to a MUSE-AstroWISE system should include the generation of data handling methods. A later stage will include deployment of software to the MUSE community. It should be noted that other CPL-based pipelines can be added with a small amount of effort.

### References


- Valentijn et. al. 2007, ASPC, 376, 491
- Weilbacher et.al. 2006, NewAR, 50, 405


Target is supported by Samenwerkingsverband Noord Nederland, Gemeente Groningen, European fund for regional development, Ministry of economic affairs, Pieken in de Delta, Provinces of Groningen and Drenthe. Target operates under the auspices of Sensor Universe.

investing in your future
European Regional Development Fund
European Union