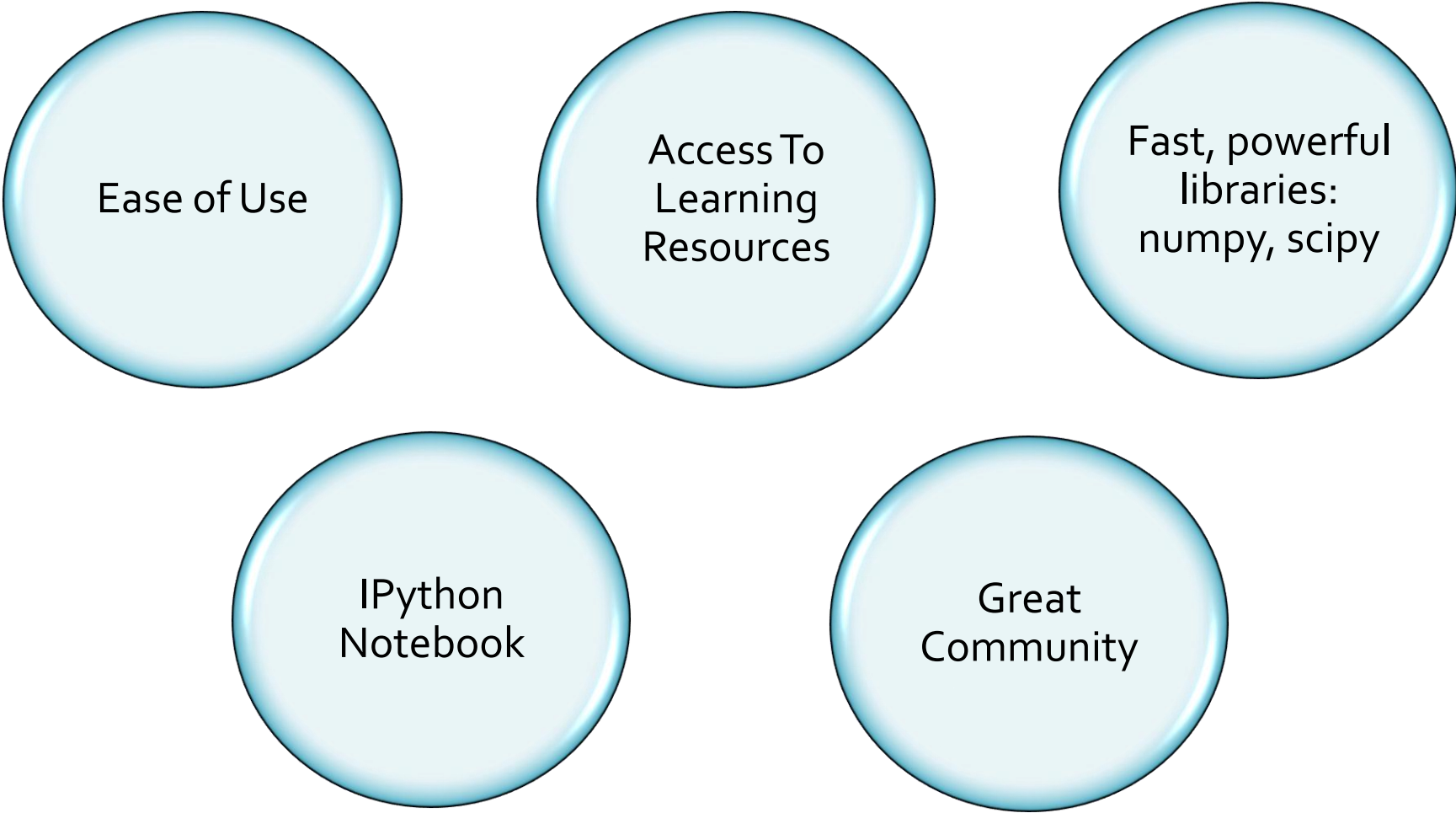


Tracy Miranda, Kichwa Coders – EclipseCon Europe 2015

Integrating Java and Python

Python for Non-Traditional Programmers



Ease of Use

Access To
Learning
Resources

Fast, powerful
libraries:
numpy, scipy

IPython
Notebook

Great
Community

JAVA

PYTHON

Statically -Typed



Dynamically-Typed

Verbose



Concise

Scales well



Ideal for beginners

Loved by tool developers
and enterprise



Loved by academics
and researchers

Eclipse RCP, SWT,
EMF



Numpy, scipy, matplotlib,
IPython notebook

JAVA + PYTHON: In Harmony Together

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

```
if __name__ == '__main__':  
    print "Hello, World!"
```



Case Study: Diamond LightSource



Scaling Scientific Tools

2005:

"I have all the data I have ever collected on a floppy disk and process it by hand..."

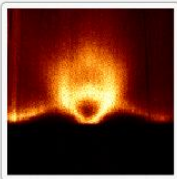

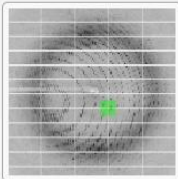
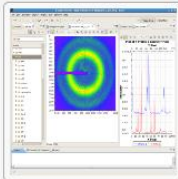
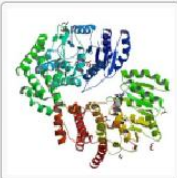
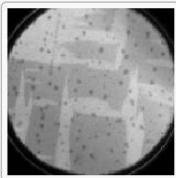

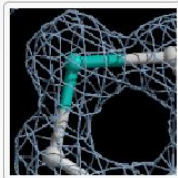
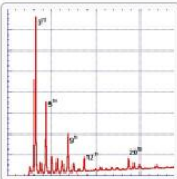
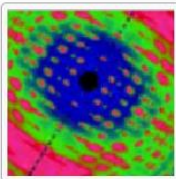
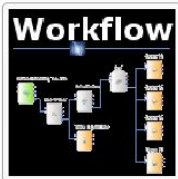
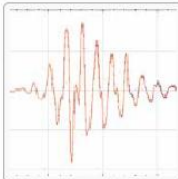
- Principal Beam Line Scientist
- Diamond Light Source Synchrotron

- (hat-tip: Mark Basham: Mark Basham - [Python for High Throughput Science](#), PyData 2014)



DExplore - Dawn
File Edit Navigate Search Project Run Window Help
Welcome

List of all **perspectives** and tools available to perform data analysis.

			
ARPES Data Reduction	DEXPLORE	DIVA	Data Browsing
			
MX Live Analysis Overview	PEEMA	Python/Jython Scripting	(Pre-release) MX pipelines
			
(Pre-release) NCD Calibration	(Pre-release) NCD Data Reduction	(Pre-release) Workflows	(Pre-release) XAFS

Copyright DAWN Collaboration

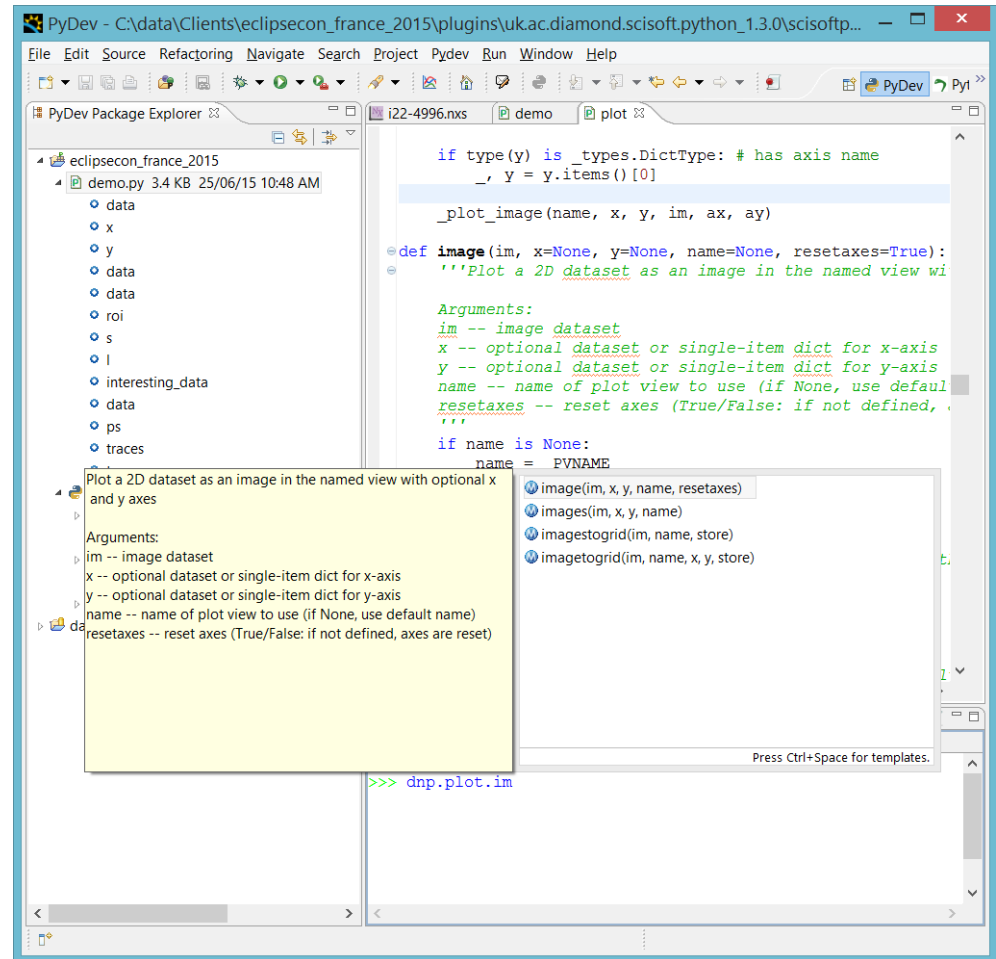
top

Requirements

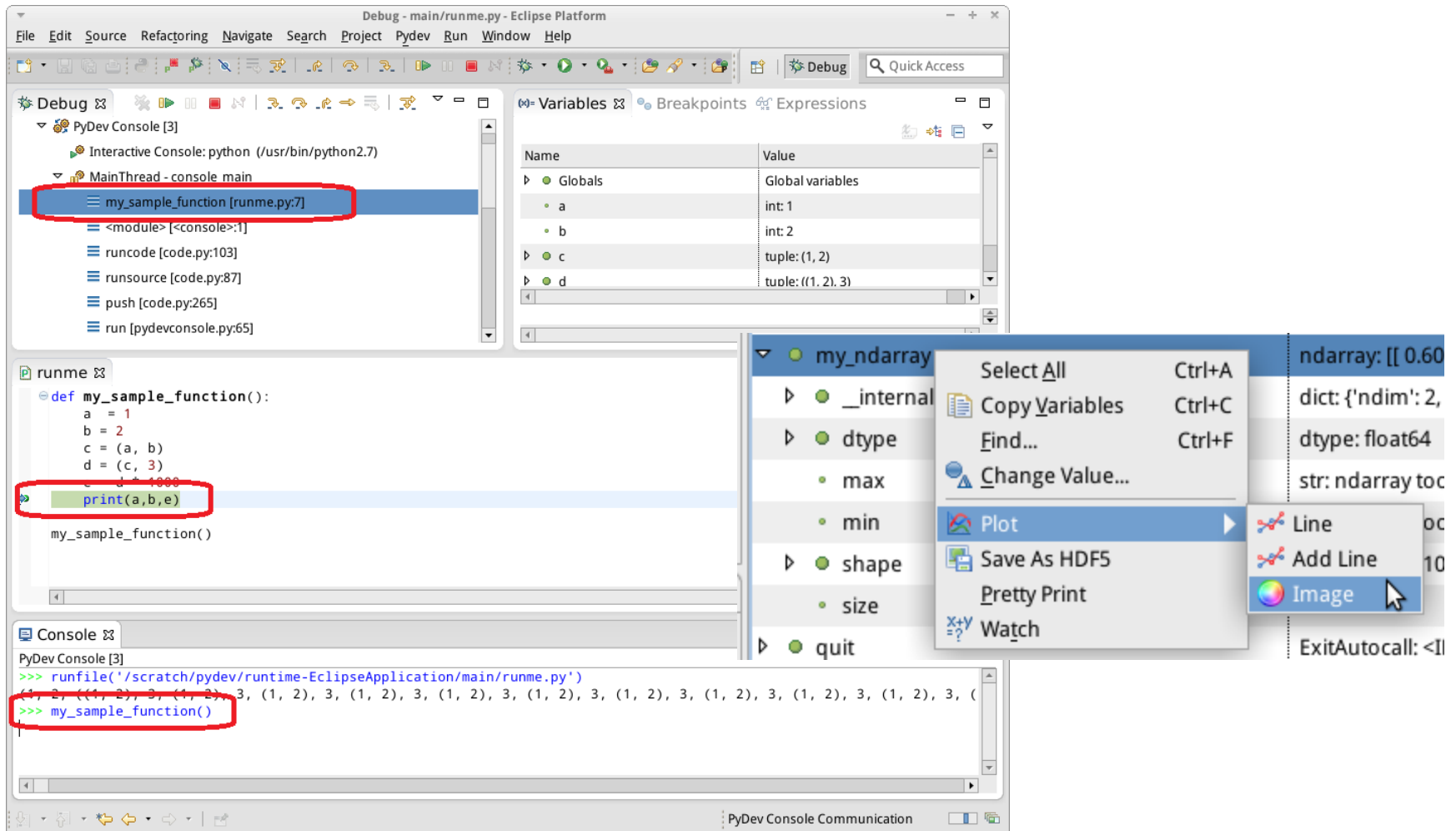
- Control the User Interface
- Python to Java
 - In Python, being able to access Java objects
 - Moving data around
 - Needed for Plotting arrays in the workbench
 - (while still running your existing scripts)
- Java to Python
 - In Java, code can access scripts in Python
 - Workflows, Extending Eclipse with Python

1. PyDev

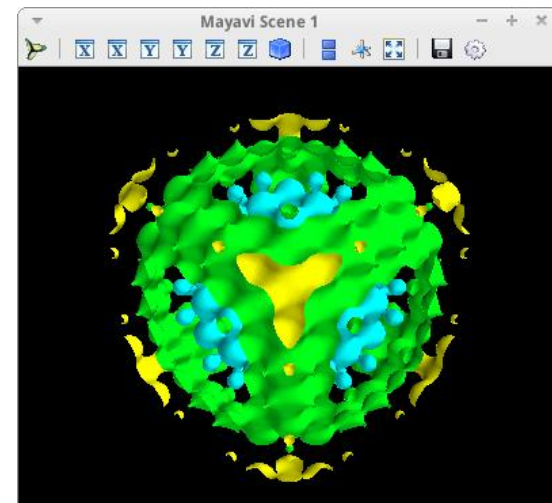
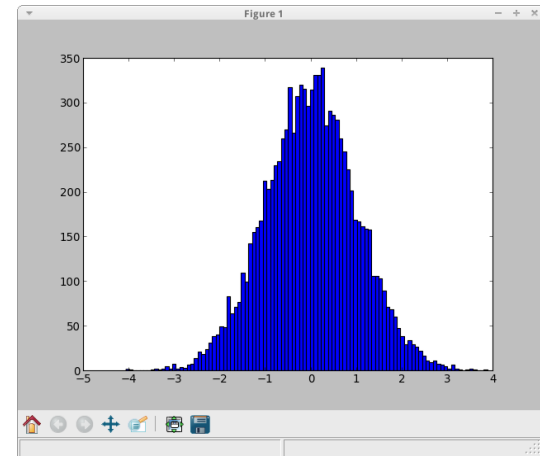
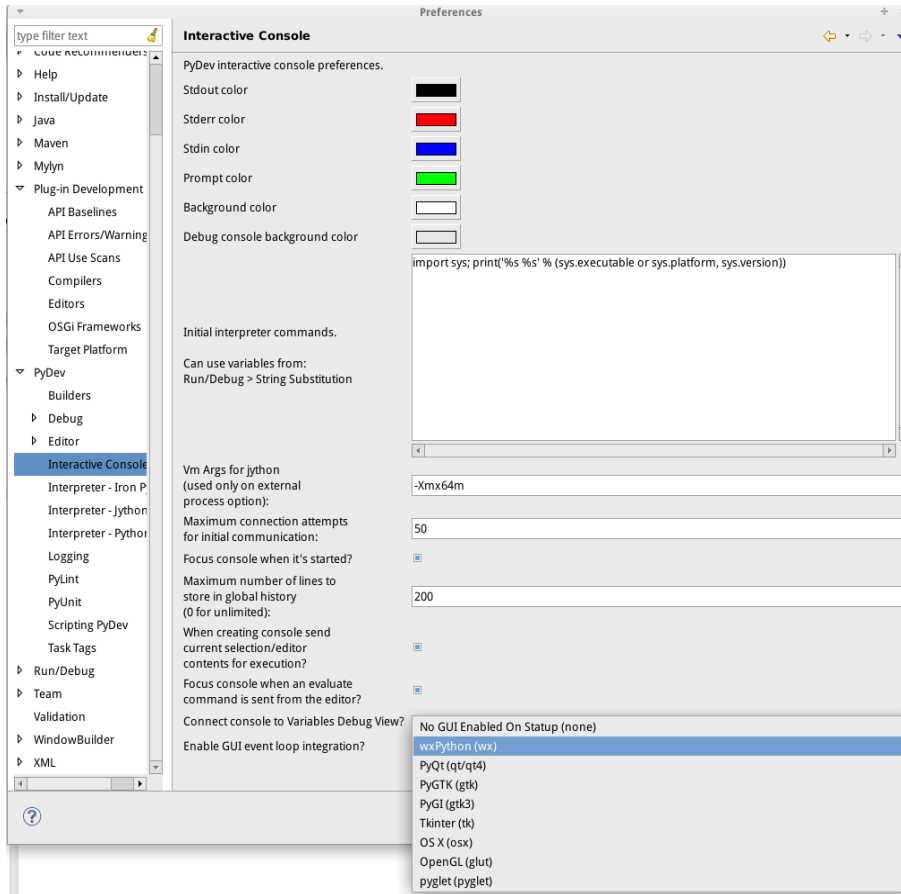
- C Python
- Jython
- PyPy
- IronPython
- Cython
- IPython



1. PyDev - Debug

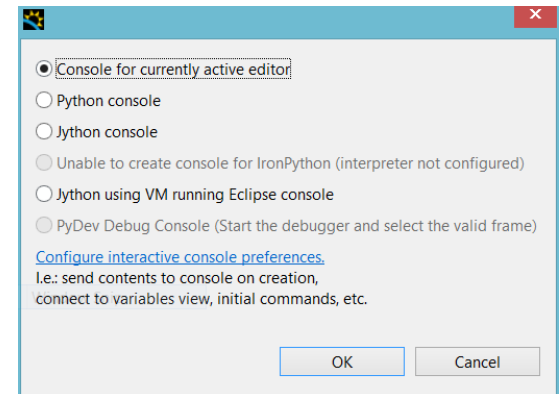


1. PyDev - Integration



2. Jython

- ☑ Java implementation of Python
- ☑ Runs in JVM so can access all JAVA classes
- ✗ Jython cannot access C modules
- ✗ JyNi could solve this, but
 - Only initial numpy support
 - No Windows support
 - Licensing issues
- ✗ Separate environment

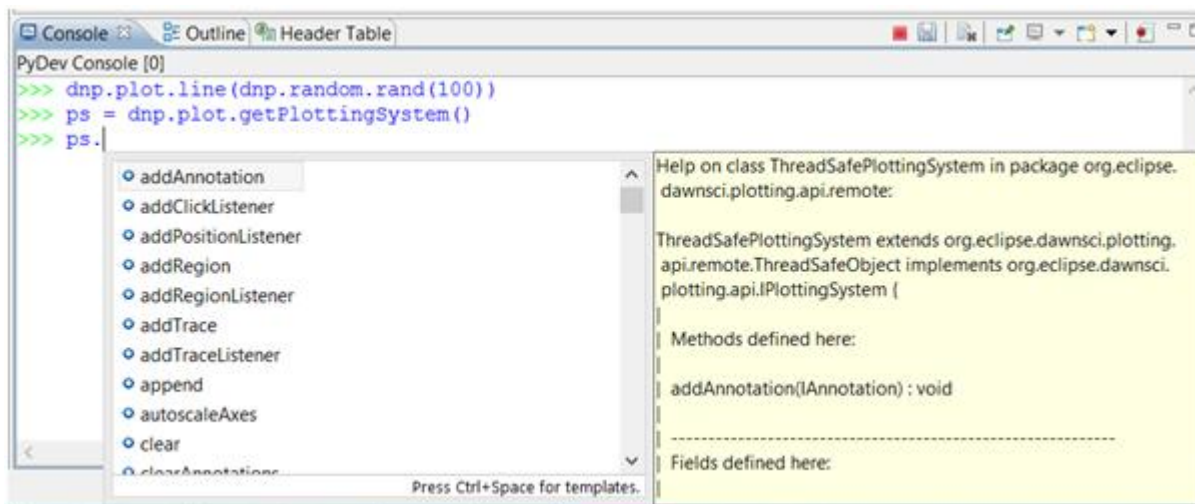


3. Py4J

- ☑ “A Bridge between Java and Python”
- ☑ Runs in JVM and allows Python to access JAVA classes
- ☑ Enables Python to Java
- ☑ Allows control of User Interface and ability to run your scripts
- Creates proxies rather than moving data
- Exposes everything
- No Java to Python support

3. Py4J

- Interactive autocompletion of Java code in Python
- Autogenerated pop-up help based on method signatures



4. AnalysisRPC

Custom solution to move data around

Enhanced Remote Procedure Call

- Datasets (ndarrays)
- Regions Of Interest (ROIs)
- Exceptions

Language Independent

Server and client written for Java and Python

Python

```
x = numpy.ndarray(...)  
y = numpy.ndarray(...)  
rpcclient.Plotter.line( x, y)
```

FLATTEN

Java

```
import  
org.eclipse.dawnsci.IDataset  
class Plotter {  
    public void line(  
        IDataset x, IDataset y) {  
        // ...  
    }  
}
```

UNFLATTEN

Transport

(XML-RPC + Disk/Memory)

Meta and simple data:
Target → Plotter
Method → line
x → /tmp/x
y → /tmp/y

Datasets:
/tmp/x
/tmp/y

Java

```
interface IFilter {  
    IDataset filter(IDataset data);  
}  
//create client for PyFilter  
client.newProxyInstance(IFilter.class)  
IDataset result = proxy.filter(data)
```

FLATTEN

Python

```
class PyFilter:  
    def filter(data):  
        # manipulate ndarray data  
        return data  
# create server  
server.add_handler(PyFilter())
```

UNFLATTEN

Transport

(XML-RPC + Disk/Memory)

Meta and simple data:
Target → PyFilter
Method → filter
data → /tmp/data

Datasets:
/tmp/data

Python To Java

■ Workflows – Python Actor

The screenshot displays the DAWN Science Workflow Editor interface. The main workspace shows a workflow diagram with the following components: a red 'Director' actor, a green 'MoKedge_1_15' actor, a blue 'Python PyDev' actor, a grey 'Data Export' actor, and an orange 'Open File' actor, all connected in a linear sequence. The 'Python PyDev' actor is selected, and its properties are visible in the bottom panel.

Project Explorer (Left):

- workflow_example_project
 - data
 - examples
 - directory_packing_example.moml 6.3 KB
 - folder_example.moml 15.7 KB 12/08/13
 - folder_monitor_example.moml 4.7 KB 1
 - gda_scan_example.moml 8.1 KB 12/08/13
 - if_example.moml 8.2 KB 12/08/13 01:38
 - loop_example.moml 12.5 KB 12/08/13 0
 - maths_example.moml 15.8 KB 12/08/13
 - maths_example2.moml 11.9 KB 12/08/13
 - maths_example3.moml 16.2 KB 12/08/13
 - motor_example.moml 11.8 KB 12/08/13
 - plot_fit_example.moml 7.8 KB 12/08/13
 - plot_image_region_example.moml 10.8 KB
 - python_numpy_example1.moml 7.2 KB
 - python_pydev_numpy_example1.moml**
 - spec_scan_example.moml 6.6 KB 12/08/13
 - user_interface_example.moml 11.3 KB 1
 - examples-edna
 - examples-icat
 - src

Properties Panel (Bottom):

Property	Value
Name	Python PyDev
Create Separate Interpreter	<input type="checkbox"/>
Dataset Outputs	InIOIt
Interpreter	workflow_example_project's PyDev interpreter settings - /usr/bin/python2.7
Pass Inputs On	<input type="checkbox"/> workflow_example_project's PyDev interpreter settings - /usr/bin/python2.7
Python Script	Default - /usr/bin/python2.7
Run Script in Debug Mode (required)	<input type="checkbox"/> python - /usr/bin/python2.7
	epd_free - /home/jonah/epd_free/bin/python2.7
	cctbx - /home/jonah/Downloads/cctbx/cctbx_plus_build/bin/cctbx.python

AnalysisRPC

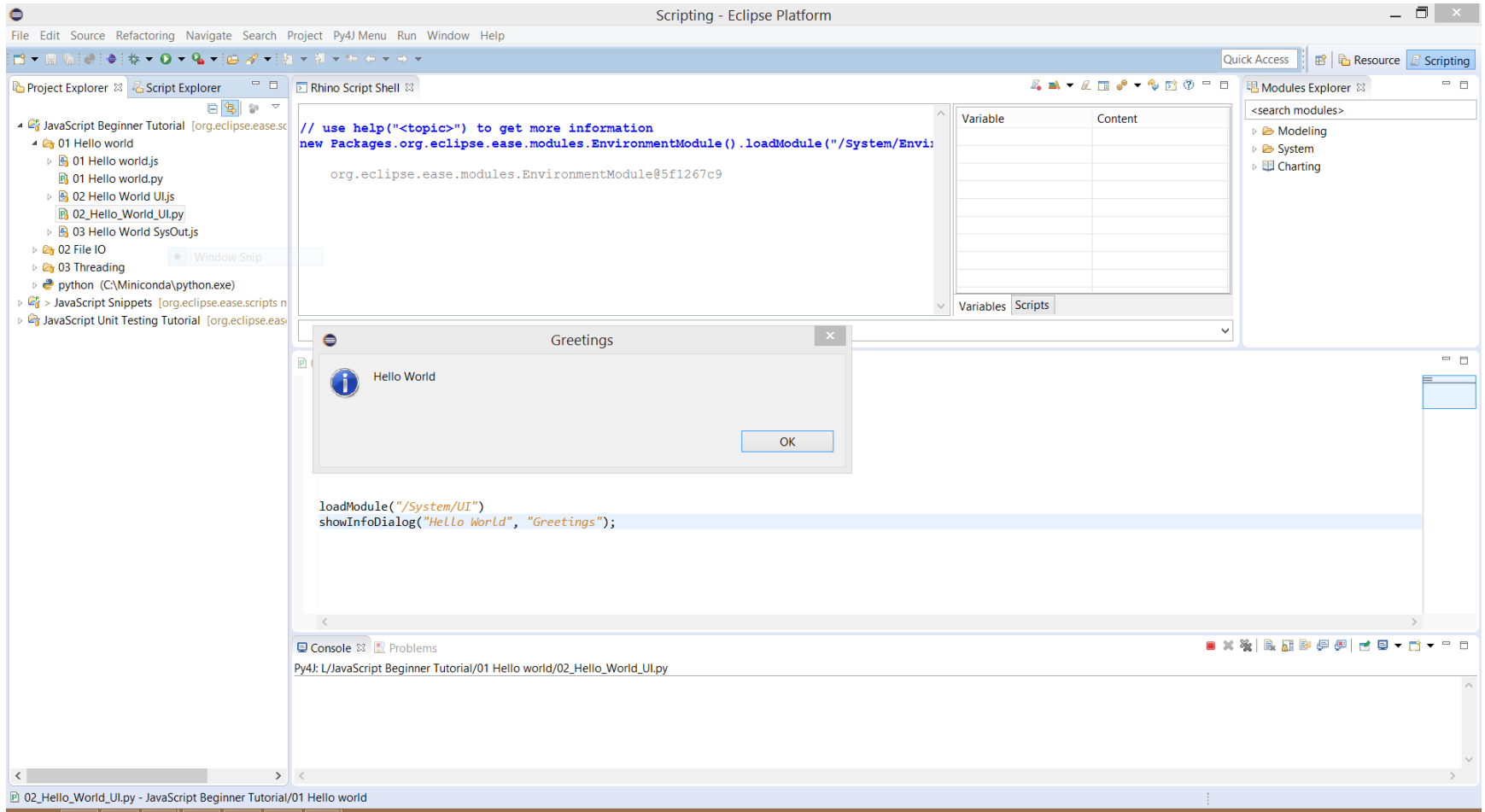
- ☑ Allows moving data around efficiently
 - ☑ Java to Python, Python to Java
 - ☑ Generic Calling Infrastructure
 - ☑ Way of Handling Exceptions
-
- ✗ In DAWNSci today – needs repackaging for use more widely.
 - ✗ Support for custom datatypes

Java & Python

Java and Python Integration is possible thanks to a mix of different technologies:

- PyDev
 - Jython
 - AnalysisRPC
 - Py4J
-
- Going forward can we unify this technology?

Eclipse Advanced Scripting Environment (EASE)



Uniting Python In Eclipse

1. Integrate Py4J into EASE
2. Integrate AnalysisRPC into EASE
3. Integrate EASE (now with Py4J and AnalysisRPC) into SWG projects

Scope?

- Support for additional data structures
- Annotations, code generation

Uniting Python in Eclipse

1. Integrate Py4J into EASE

- Py4JPythonScriptEngine
 - Threading model
 - Memory Management
- User Experience
 - Integrate best of PyDev
- What about EASE into Eclipse Platform?

Python & Java

With tight integration Java & Python work together to create a powerful environment for Science, and beyond!

Keep upto date with latest at science working group forum

science.eclipse.org



Questions?

- Tracy Miranda tracy@kichwacoders.com
- Jonah Graham



n