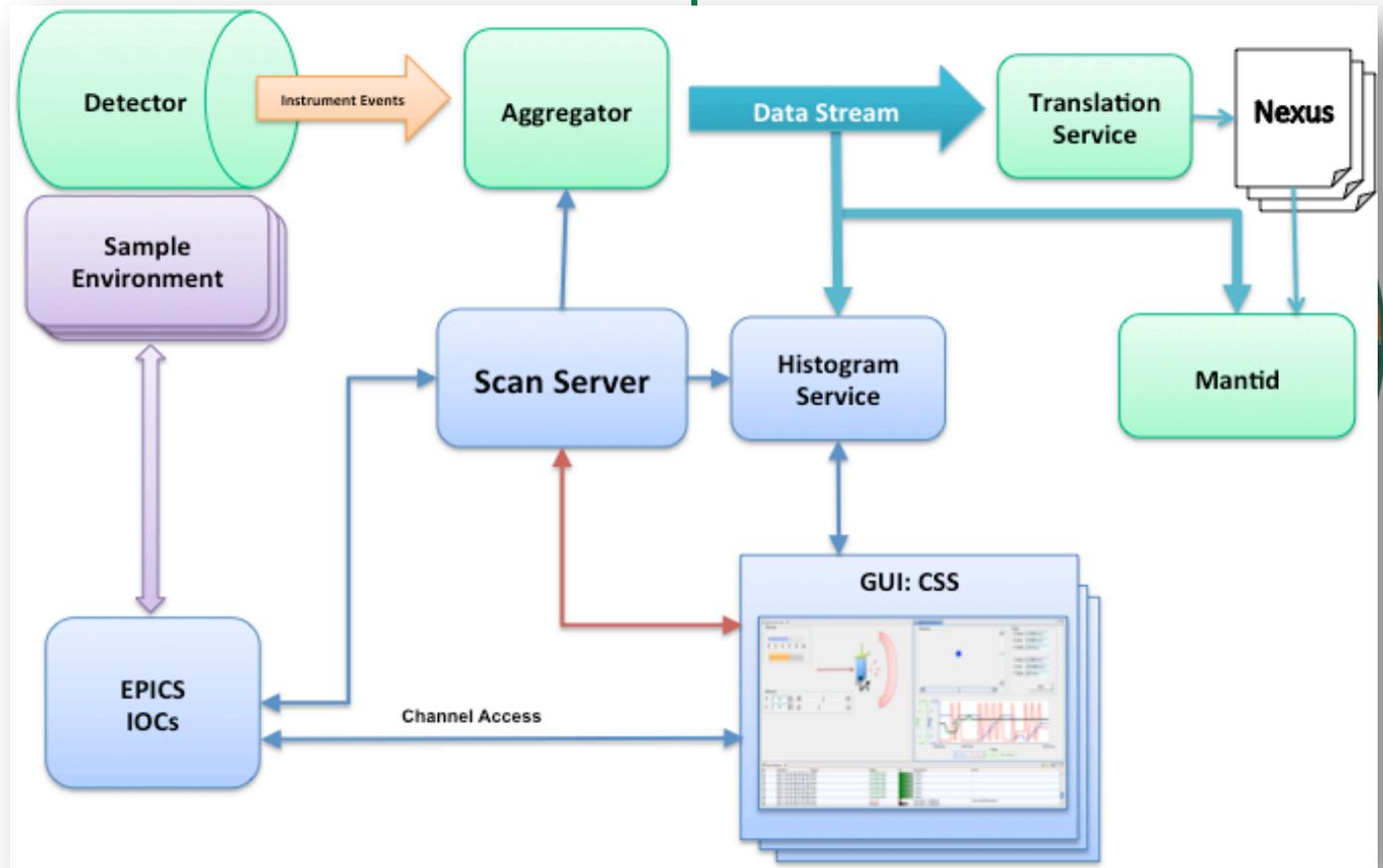


Scan System

Kay Kasemir,
Xihui Chen

Jan. 2013



Automated Experiment Control

“Scan” should be easy

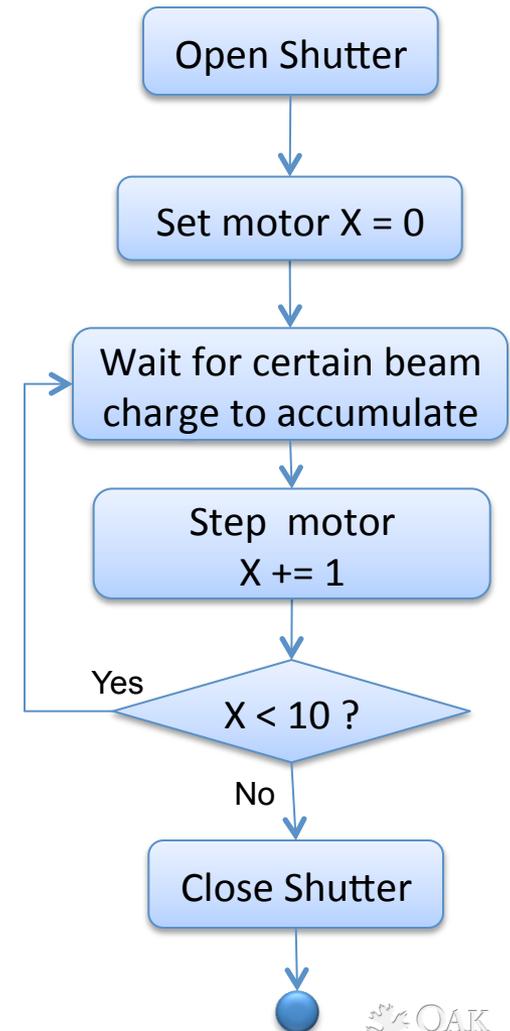
Graphical:

Loop 'xpos' = 0.0 ... 10.0, step 1.0 (wait for 'xpos' +/-0.1)
Wait for 'pcharge' to increase by 3.0E12

... or script:

```
scan(('xpos', 1, 10),  
     WaitCommand('pcharge', Comparison.INCREASE_BY, 3e12)  
     )
```

- Robust execution
- Monitor, pause, resume, abort



EPICS Sequencer?

Excellent for “permanent” automation,
not for changing scan scenarios

- **Compile, link, ...**
- **No GUI**
- **No progress, pause, resume**
- **Can do anything**

Scripts (Python, Jython, Scala, ...)

“Works”, but...

- No progress, pause, resume
- Can do most anything

Allow Anything

≠

Robust

```
set(setpoint, 30);  
# Assume OK after 10 seconds  
wait(10.0);
```

```
set(nonexisting_channel, 42);
```

```
while (readback < 10)  
{ /* busy loop */ }
```

```
set(setpoint, 30);  
# Wait for readback to match  
waitForValue(readback, 30);
```

```
if (notConnected(...))  
    reportError();
```

```
while (readback < 10)  
{  
    sleep(1);  
    if (timeout_exceeded)  
        reportError();  
}
```

APS “Scan Engine”

Claude Saunders, Mitch McCuiston, Brian Tieman, Tim Mooney

✓ “Scan Engine” executes submitted scans

✓ Scan = List of robust commands

“Loop” command

Pause, Resume

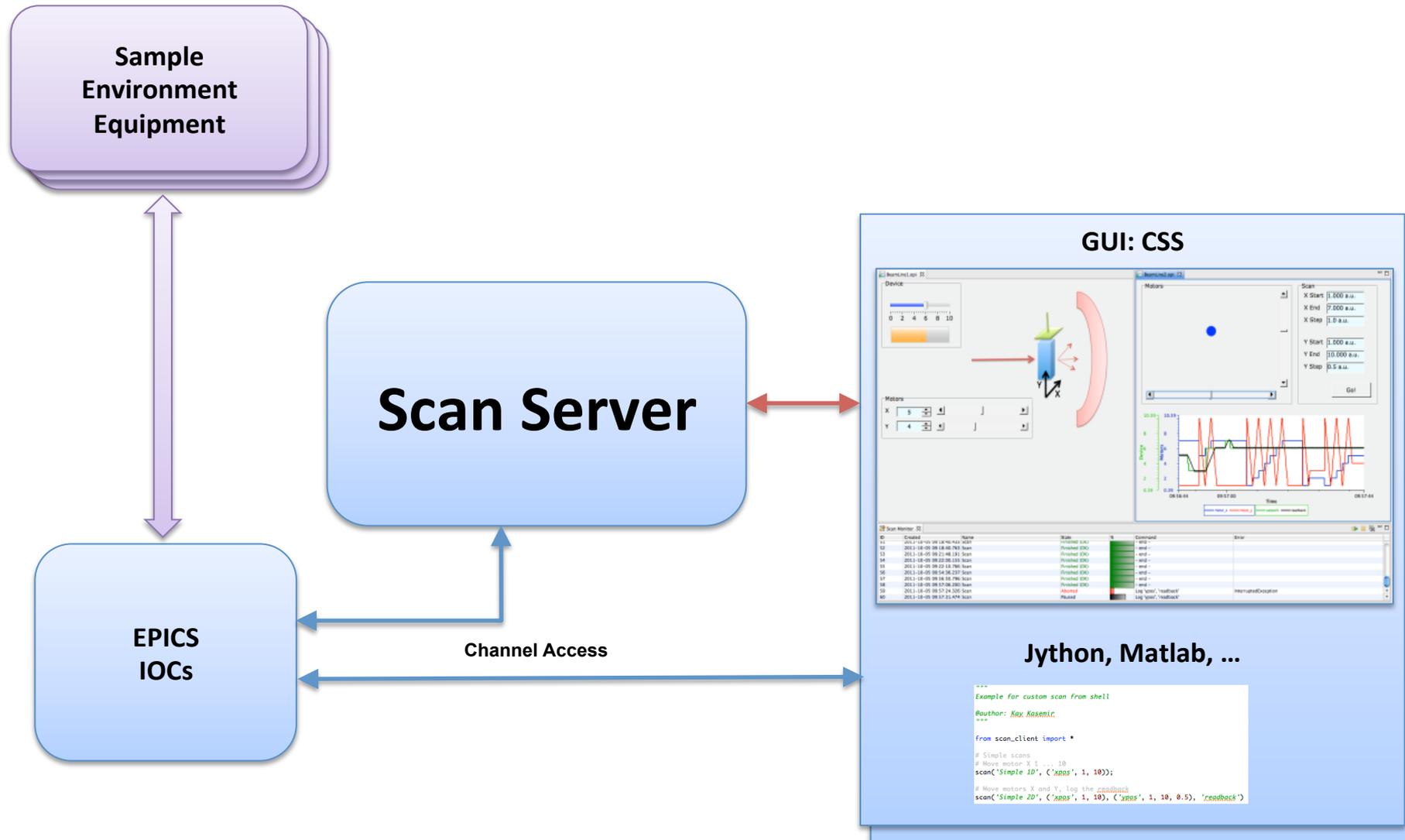
Eclipse RCP instead of Spring

Eclipse build instead of Maven

CSS PV instead of new PV layer

Tight CSS GUI integration AND basic script access

Experiment Control for EPICS



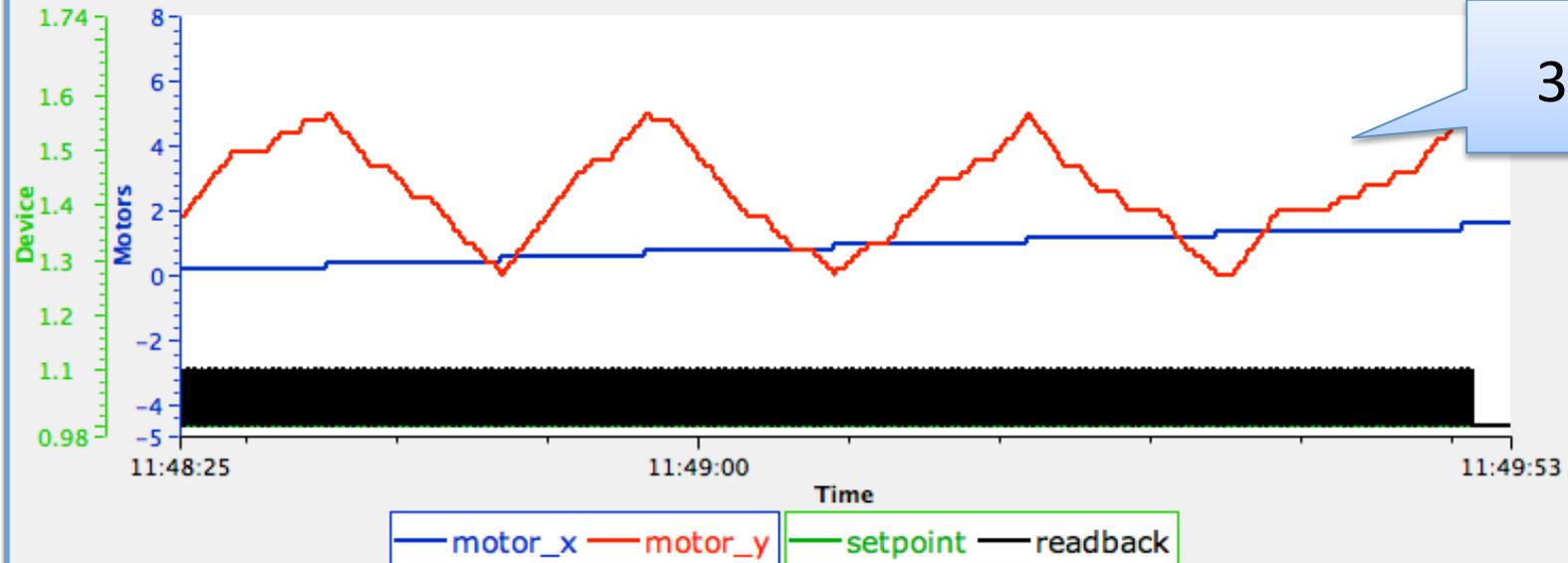
BOY Example

1. Configure

The interface is divided into three main sections:

- Beam:** Contains a green circle icon for the beam, a blue square icon for the shutter, and a text box showing "Neutrons 336,338".
- Motors:** A large empty area with a blue circle in the center and navigation arrows.
- Scan:** A configuration panel with the following fields:
 - X: Start (0.0 a.u.), End (5.0 a.u.), Step (0.200 a.u.)
 - Y: Start (0.0 a.u.), End (5.0 a.u.), Step (0.200 a.u.)
 - Neutrons: 3.0 a.u.
 - Name: XY Scan2
 - Up/Down: A green slider.
 - Active: A green circle.
 - Go!: A button with a green arrow.

2. Start



3. Monitor

Submit Scan from Table Example

Point by Point Scan

Nested Scan

Points	xpos	ypos	setpoint
Point 1	0	0	5
Point 2	1	1	10
Point 3	2	2	15
Point 4	3	3	20
Point 6	4	4	15
Point 7	5	5	10
Point 8	6	6	0

Submit Scan

 in workspace

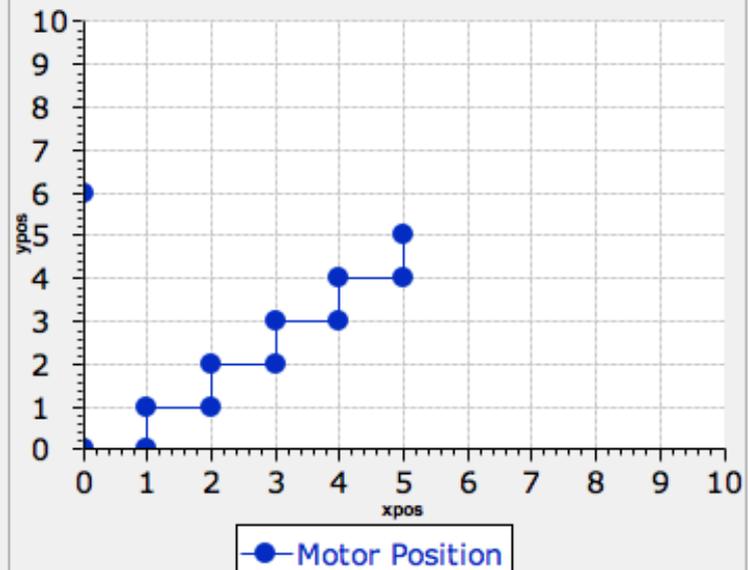
Load from .csv ...

Export to .csv file

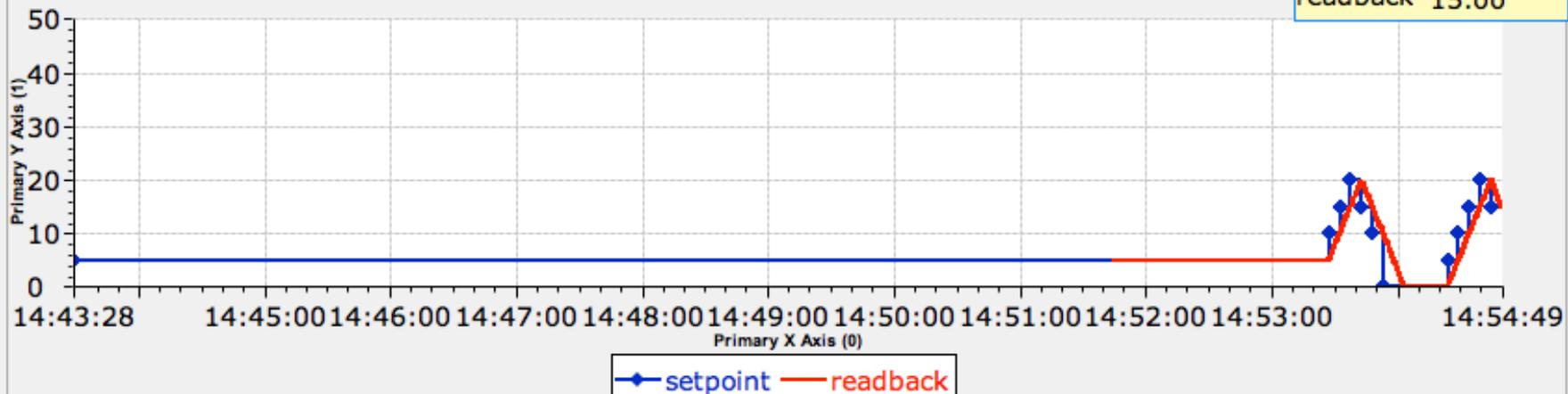
Scan

Running

Motor Trace



setpoint: 10
readback 15.00



Current Running

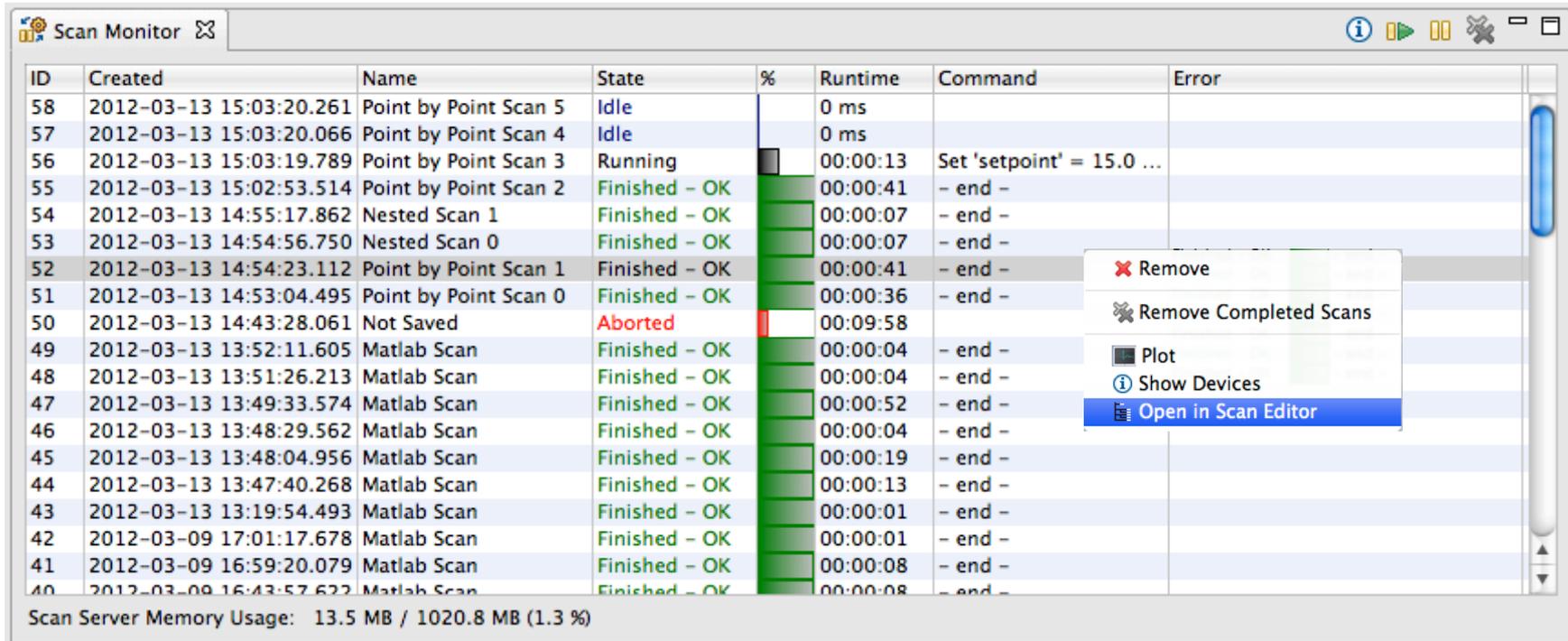
Point by Point Scan

Current

Set 'ypos' = 5.0 (wait for

79%

Scan Monitor



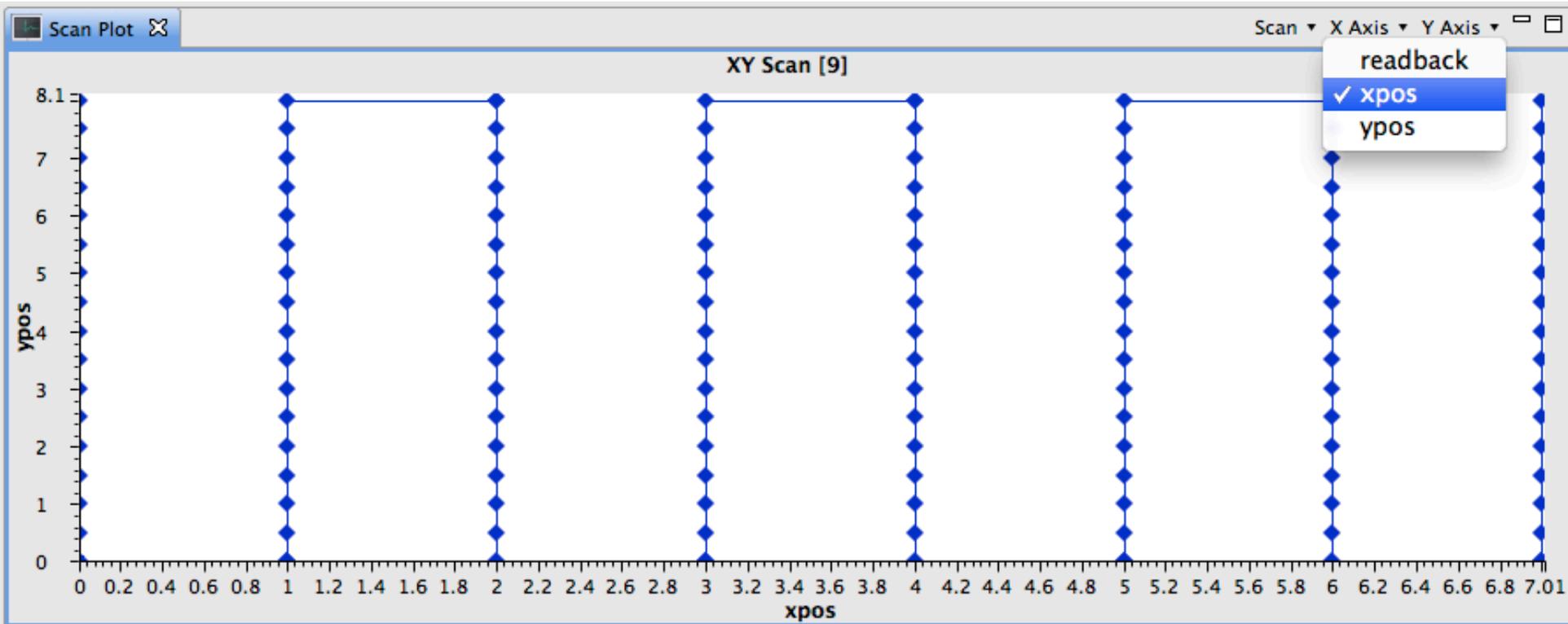
ID	Created	Name	State	%	Runtime	Command	Error
58	2012-03-13 15:03:20.261	Point by Point Scan 5	Idle		0 ms		
57	2012-03-13 15:03:20.066	Point by Point Scan 4	Idle		0 ms		
56	2012-03-13 15:03:19.789	Point by Point Scan 3	Running		00:00:13	Set 'setpoint' = 15.0 ...	
55	2012-03-13 15:02:53.514	Point by Point Scan 2	Finished - OK		00:00:41	- end -	
54	2012-03-13 14:55:17.862	Nested Scan 1	Finished - OK		00:00:07	- end -	
53	2012-03-13 14:54:56.750	Nested Scan 0	Finished - OK		00:00:07	- end -	
52	2012-03-13 14:54:23.112	Point by Point Scan 1	Finished - OK		00:00:41	- end -	
51	2012-03-13 14:53:04.495	Point by Point Scan 0	Finished - OK		00:00:36	- end -	
50	2012-03-13 14:43:28.061	Not Saved	Aborted		00:09:58		
49	2012-03-13 13:52:11.605	Matlab Scan	Finished - OK		00:00:04	- end -	
48	2012-03-13 13:51:26.213	Matlab Scan	Finished - OK		00:00:04	- end -	
47	2012-03-13 13:49:33.574	Matlab Scan	Finished - OK		00:00:52	- end -	
46	2012-03-13 13:48:29.562	Matlab Scan	Finished - OK		00:00:04	- end -	
45	2012-03-13 13:48:04.956	Matlab Scan	Finished - OK		00:00:19	- end -	
44	2012-03-13 13:47:40.268	Matlab Scan	Finished - OK		00:00:13	- end -	
43	2012-03-13 13:19:54.493	Matlab Scan	Finished - OK		00:00:01	- end -	
42	2012-03-09 17:01:17.678	Matlab Scan	Finished - OK		00:00:01	- end -	
41	2012-03-09 16:59:20.079	Matlab Scan	Finished - OK		00:00:08	- end -	
40	2012-03-09 16:43:57.622	Matlab Scan	Finished - OK		00:00:08	- end -	

Scan Server Memory Usage: 13.5 MB / 1020.8 MB (1.3 %)

List Scans on Server

- Idle: To be executed next
- Running: With progress report
- Finished, Failed: Past runs

Scan Plot



- Plot variables used by scan
- Get data from Running or Finished scans

Scan Editor

Open, save

Add commands

Undo ⌘Z
Redo ⇧⌘Z
Cut ⌘X
Copy ⌘C
Paste ⌘V
Delete ⌘X
Simulate Scan
Submit Scan
Add Command
Open Property Editor
Open Command Palette
Open/Reset Scan Editor Perspective
Process Variable

Scan Command Palette

This is a comment
Delay 1.0 sec
Log 'device'
Loop 'device' = 0.0 ... 10.0, step 1.0 (wait for 'd'
x = Set 'device' = 0.0 (wait for 'device' +-0.1)
Wait for 'device' = 0.0 (+-0.1)

Properties

Property	Value
1. Device Name	pcharge
2. Comparison	to increase by
3. Desired Value	1.0E12
4. Tolerance	0.1
5. Time out (secon...	0.0

Set parameters

Not logged in

- “Undo”
- Drag/drop commands or PV names (also as XML text)
- Device PVs (or alias) can be picked from beamline-specific configuration

Simulation Mode

The left window, titled 'powgen.scn', displays a list of simulation steps. Each step consists of a setpoint or wait command followed by a value. The values are: 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, and 0.0. The wait commands are for 'pcharge' to increase by 3.5E12 or 1.75E12. The right window, titled 'Scan Simulation', shows a log of simulation events. Each event is a timestamp followed by a setpoint or wait command and a value in brackets. The values are: 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, and 0.0. Red circles highlight the values 1.0 and 0.0 in the right window.

- Simulated PV changes
- Estimates times

```
<!-- Simulation information for a PV
Slew rate is in [units of the device] / second

The (simulated) neutrons change at about 7/sec
-->
<pv>
  <name>neutrons</name>
  <slew_rate>7</slew_rate>
</pv>
<!-- Proton charge: About 3.5e12 / hour = 9.7e8 / sec -->
<pv>
  <name>pcharge</name>
  <slew_rate>9.7222e+08</slew_rate>
</pv>
```

Monitor, Adjust Live Scan

Scan 'Not Saved' [50]: Paused, 17% done

× = Set 'xpos' = 0.0 (wait for 'xpos' +-0.1)
× = Set 'ypos' = 0.0 (wait for 'ypos' +-0.1)
× = **Set 'setpoint' = 5.0 (wait for 'readback' +-0.1)**
Log 'readback'
× = Set 'xpos' = 1.0 (wait for 'xpos' +-0.1)
× = Set 'ypos' = 1.0 (wait for 'ypos' +-0.1)
× = Set 'setpoint' = 10.0 (wait for 'readback' +-0.1)
Log 'readback'
× = Set 'xpos' = 2.0 (wait for 'xpos' +-0.1)
× = Set 'ypos' = 2.0 (wait for 'ypos' +-0.1)
× = Set 'setpoint' = 15.0 (wait for 'readback' +-0.1)
Log 'readback'
× = Set 'xpos' = 3.0 (wait for 'xpos' +-0.1)
× = Set 'ypos' = 3.0 (wait for 'ypos' +-0.1)
× = Set 'setpoint' = 20.0 (wait for 'readback' +-0.1)
Log 'readback'
× = Set 'xpos' = 4.0 (wait for 'xpos' +-0.1)
× = Set 'ypos' = 4.0 (wait for 'ypos' +-0.1)
× = Set 'setpoint' = 15.0 (wait for 'readback' +-0.1)
Log 'readback'
× = Set 'xpos' = 5.0 (wait for 'xpos' +-0.1)
× = Set 'ypos' = 5.0 (wait for 'ypos' +-0.1)
× = Set 'setpoint' = 10.0 (wait for 'readback' +-0.1)
Log 'readback'
× = Set 'xpos' = 6.0 (wait for 'xpos' +-0.1)
× = Set 'ypos' = 6.0 (wait for 'ypos' +-0.1)
× = Set 'setpoint' = 0.0 (wait for 'readback' +-0.1)
Log 'readback'

Scan Info

Delay 1.0 sec
Log 'device'
Loop 'device' = 0.0 ... 10.0, step 1.0 (wait for 'device' +-0.1)
× = Set 'setpoint' = 5.0 (wait for 'readback' +-0.1)

Active Command

Properties

Property	Value
▼ Set	
1. Device Name	xpos
2. Value	2.0
3. Wait for readback	yes
4. Readback Device	xpos
5. Tolerance	active
6. Time out (seconds; 0 to disable)	ypos

Adjust properties of live scan

Scripted Scan

Jython editor,
debugger

Jython console

```
*custom_scan
"""Example for custom scan from shell"""

from scan_client import *

# Simple scans
# Move motor X 1 ... 10
scan('Simple 1D', ('xpos', 1, 10));

# Move motors X and Y, log the readback
scan('Simple 2D', ('xpos', 1, 10), ('ypos', 1, 10, 0.5), 'readback')

# Connect to server for more detailed custom scans
client = ScanClient()
cmds = [
    DelayCommand(2.0),
    LoopCommand('xpos', 1, 5, 0.1,
        [
            SetCommand('setpoint', 1),
            WaitCommand('readback', Comparison.EQUALS, 1.0, 0.5, 0.0),
            SetCommand('setpoint', 5),
            WaitCommand('readback', Comparison.EQUALS, 3.0, 0.5, 0.0),
            LogCommand(['xpos', 'readback'])
        ]
    ),
]

# Schedule for execution on server
seq = CommandSequence(
seq.dump()
```

- Run As
- Debug As
- Team
- Compare With
- Replace With
- PyDev
- 1 Jython Run
- 2 Jython unit-test
- 3 Python Run
- 4 Python unit-test
- Run Configurations...

```
Console
PyDev Console [0]
>>> import sys;
>>> print('%s %s' % (sys.executable or sys.platform, sys.version))
java1.6.0_29 2.5.2 (Release_2_5_2:7206, Mar 2 2011, 23:12:06)
[Java HotSpot(TM) 64-Bit Server VM (Apple Inc.)]
>>>
>>> sys.path.append('/Kram/MerurialRepos/cs-studio/products/SNS/plugins/org.csstudio.scan/examples')
>>> from scan_client import *
>>> scan(('xpos', 1, 10))
10L
>>>
```

N-dimensional scan that logs arbitrary number of readings based on nested loops.

Arguments:

- * Optional scan name
- * One or more scan specifications: ('device', start, end[, step])
- * Names of device to log in addition to loop'ed devices

Examples:

```
# Scan 'xpos' from 1 to 10, stepping 1, automatically logging 'xpos'
```

... or use 'vi', shell

Matlab



```
Editor - /Kram/MerurialRepos/cs
```

15 %%
16 % Number of points
17 N = 20;
18 % Center, diameter
19 C = 5;
20 D = 4;
21 % Create circle
22 i = 1:N;
23 x = C + D*cos(2*pi*i/(N-1));
24 y = C + D*sin(2*pi*i/(N-1));
25 plot(x, y, '.');
26 xlim([0 10]);
27 ylim([0 10]);
28
29 %% Create a scan
30 % Scan x/y, at each step waiting for readback to follow setpoint
31
32 seq = CommandSequence();
33 for i = 1:N
34 seq.set('xpos', x(i));
35 seq.set('ypos', y(i));
36 end
37 seq.dump();
38
39 %% Submit to server
40 id = server.submitScan('Matlab Scan', seq.getXML());
41
42 %% Wait for scan to finish
43 while 1
44 info = server.getScanInfo(id)
45 scandata=server.getScanData(id);
46 scandata.getDevices();
47
48

Create Scan

Monitor

```
Command Window
```

Scan 'Matlab Scan' [49]: Finished - OK, 104% done
fx >>

script

Analyze
(maybe submit follow-up scan)

Scan Commands

- **Set**
 - Set device (PV) to a value.
 - Optional wait for read-back, same or other PV, with timeout.
- **Wait**
 - Wait until a device (PV) reaches a certain value. Support condition: $>$, $<$, $==$, $>=$, $<=$, increment-by, decrease-by. Optional timeout.
- **Loop**
 - Command that performs a loop, optional read-back and timeout.
- **Log**
 - Log data for plot in addition to variables used by Set, Wait, Loop
- **Delay**
 - Delay for a certain time. *Discouraged. Use Wait.*
- **Script**
 - Execute jython code. *Use with care.*

Custom commands can be added via Eclipse extension points.

Performance

- **Command Execution**
 - **80000 commands/second: Delay 0 sec, Set w/o read-back**
 - **4500 commands/second: Set w/ read-back, Loop**

- **Download scan into Editor**
 - **10000 commands: 1 second**
 - **50000 commands: 15 seconds**

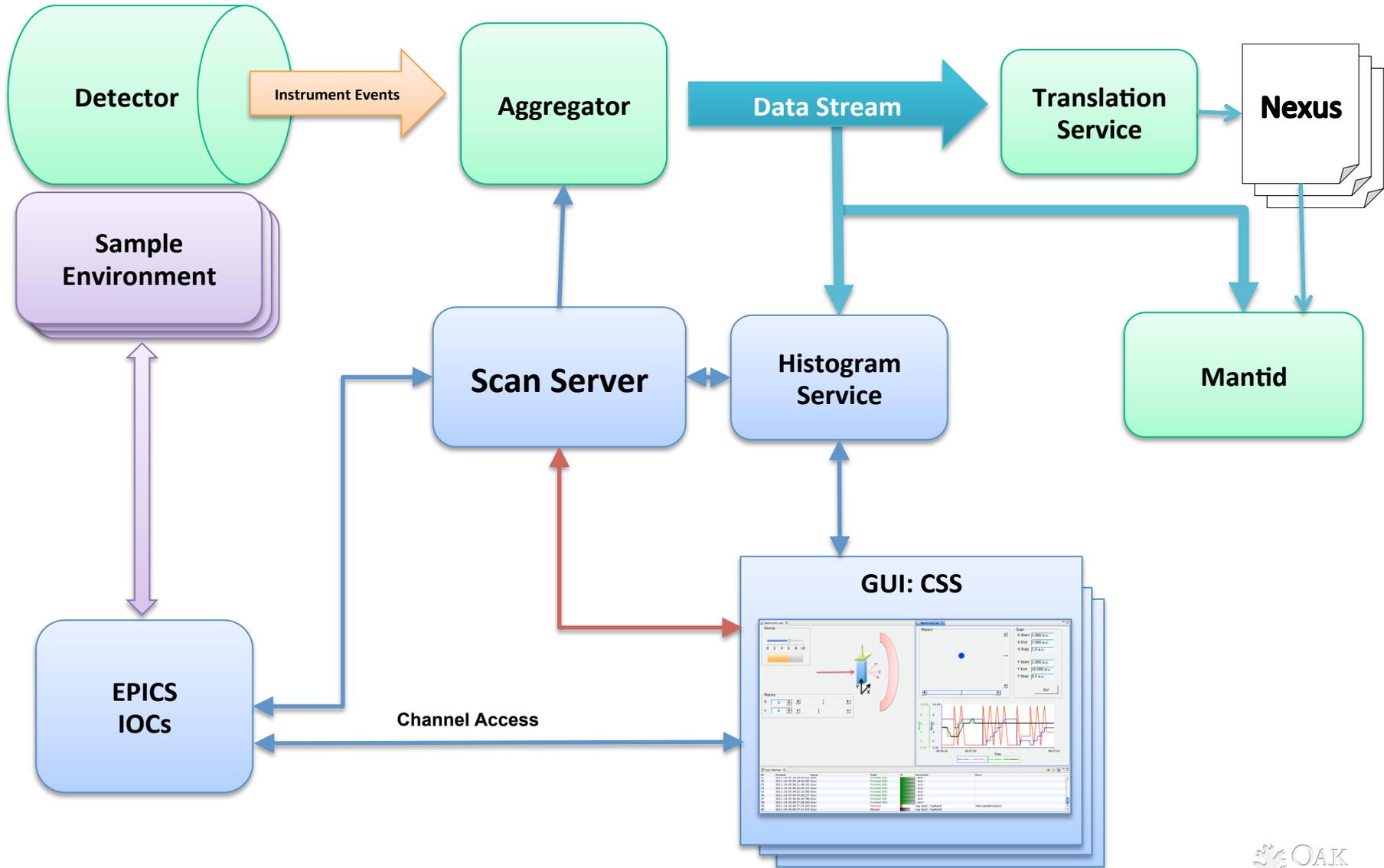
Beam Line Configuration

- Required (soft) IOCs
 - Used by separate tools to start/stop as required
- Scan System Aliases
 - Available within scans
- Simulation Info
 - Slew rates
- DAQ info
 - Which channels to log with neutron data?

```
<device>
  <name>2-D Translation Table</name>
  <!-- IOC that handles this device:
        Name,
        script that starts the soft IOC,
        console port for procServ
  -->
  <ioc>
    <name>Simulation</name>
    <script>/home/epics/R3.14.12.2/applications/bl99/simulation/st.cmd</script>
    <console>4812</console>
  </ioc>

  <!-- PVs provided by the soft IOC that are of interest
        to Scan Server, Aggregator, maybe more:
        Name, optional alias, scannable? loggable?
  -->
  <pv><name>motor_x</name><alias>xpos</alias><log/><scan/></pv>
  <pv><name>motor_y</name><alias>ypos</alias><log/><scan/></pv>
</device>
```

Overall Picture for SNS



Summary

Scan Server: Experiment Automation

- BOY Panels, Scan Editor, Scripts
- Monitor, Pause, Resume, even adjust

The screenshot displays the Scan Server interface with several components:

- Script Editor:** A list of commands including 'Set 'setpoint' = 1.0', 'Wait for 'readback' to reach 1.0 (+-0.1)', 'Delay 5.0 sec', and two loops for 'xpos' and 'ypos' with associated 'Log 'readback'' commands.
- Scan Command Palette:** A list of commands including 'Delay 1.0 sec', 'Log 'device'', 'Loop 'device' = 1.0 ... 10.0, step 1.0', 'Set 'device' = 1.0', and 'Wait for 'device' to reach 1.0 (+-0.1)'. The 'Set 'device' = 1.0' command is currently selected.
- Properties:** A table showing the current state of the selected command:

Property	Value
Set	
1. Device	setpoint
2. Value	1.0
- Scan Monitor:** A table showing the status of recent scans:

ID	Created	Name	State	%	Command	Error
3	2011-12-28 16:03:33.701	demo	Finished - OK	100%	- end -	
2	2011-12-28 15:57:03.297	demo	Failed	0%	Log 'device'	Unknown ..
1	2011-12-28 10:40:49.804	XY Scan	Finished - OK	100%	- end -	

Scan Server

- Submitted scans are queued for execution
 - Pre- and Post-Scan commands: open/close shutter, ...
- CSS/Eclipse 'headless' application
- 'PV': EPICS, simulated
 - Alias names for PVs as well as plain PV names
- Telnet interface: Status, pause, resume, ...
- Java RMI interface: Submit, status, pause, resume, get data, ...
 - Scans transferred in XML format
 - RMI = Java, Jython, Matlab, Scala, JRuby, ...

Plans, Ideas

- **RESTful web interface**

- **Submit scan**
- **Monitor scans**
- **Abort scan**

Allows non-Java tools to ‘close the loop’