



# JAVA TOOL FRAMEWORK FOR AUTOMATION OF HARDWARE COMMISSIONING AND MAINTENANCE PROCEDURES

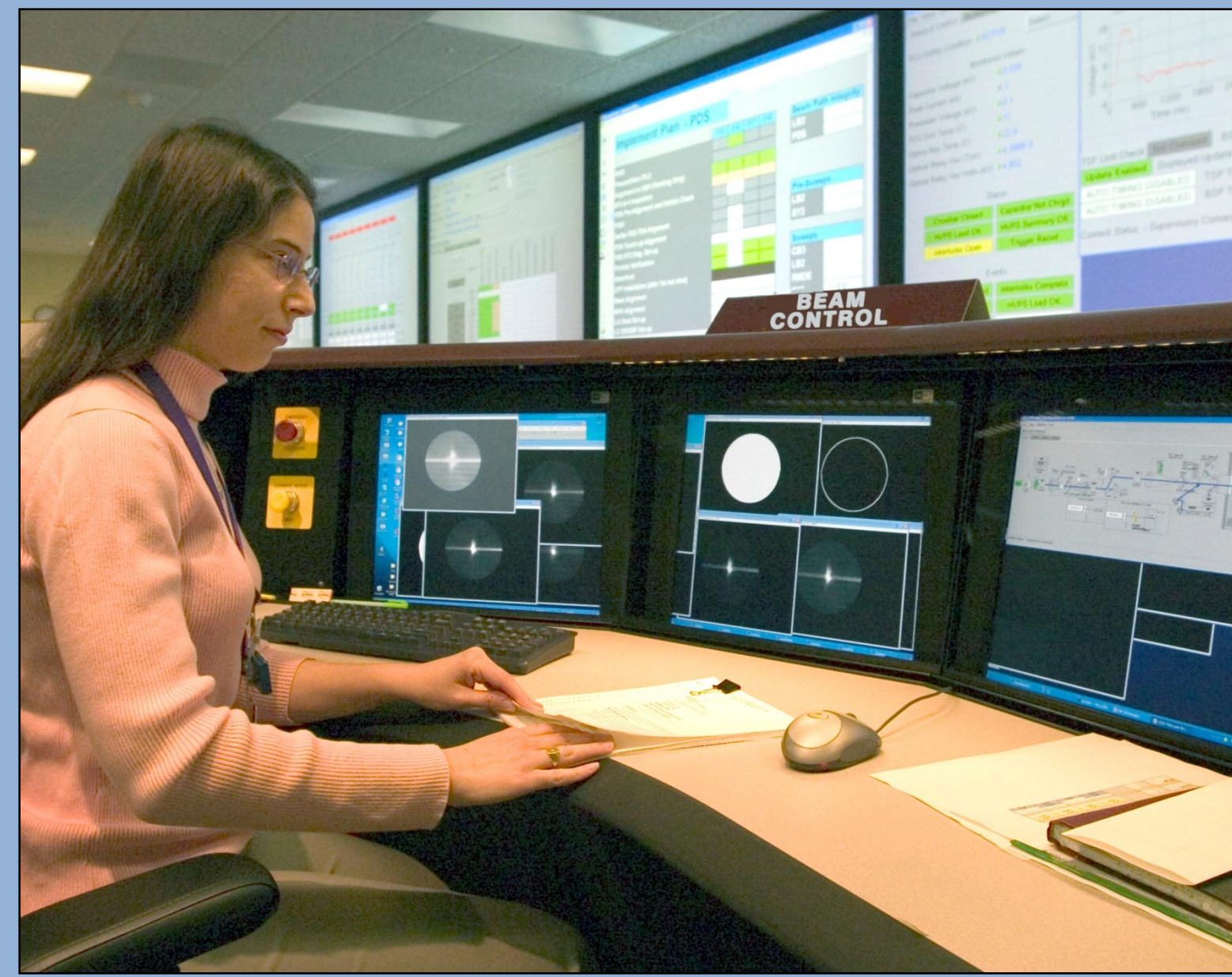


J. Ho, J. Fisher, S. West, J. Gordon, L. Lagin  
Lawrence Livermore National Laboratory, CA, U.S.A

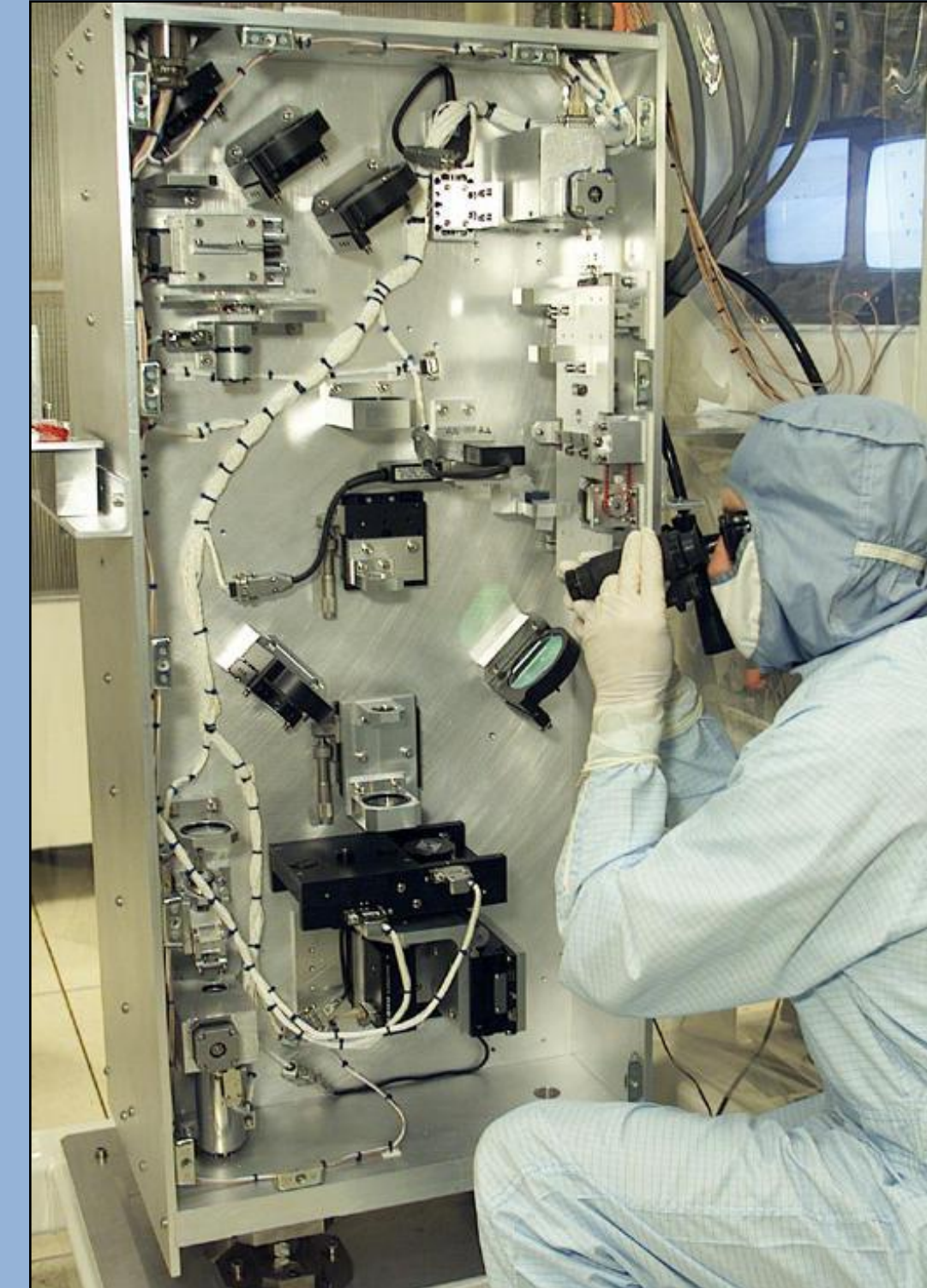
The National Ignition Facility (NIF) is a 192-beam laser system, housing 6,200 line replaceable units (LRU) that contain optics, stepping motors, and sensors to control and diagnose the laser. Each LRU must undergo qualification procedures to verify and calibrate the hardware necessary to commission and maintain the unit. Manual qualifications of the LRU are labor-intensive and error prone because an operator repeats a tedious process many times. Maintenance and Commissioning Tool software automates qualification procedures to improve the efficiency.

## MCT Overview – Automating the Qualification Procedures

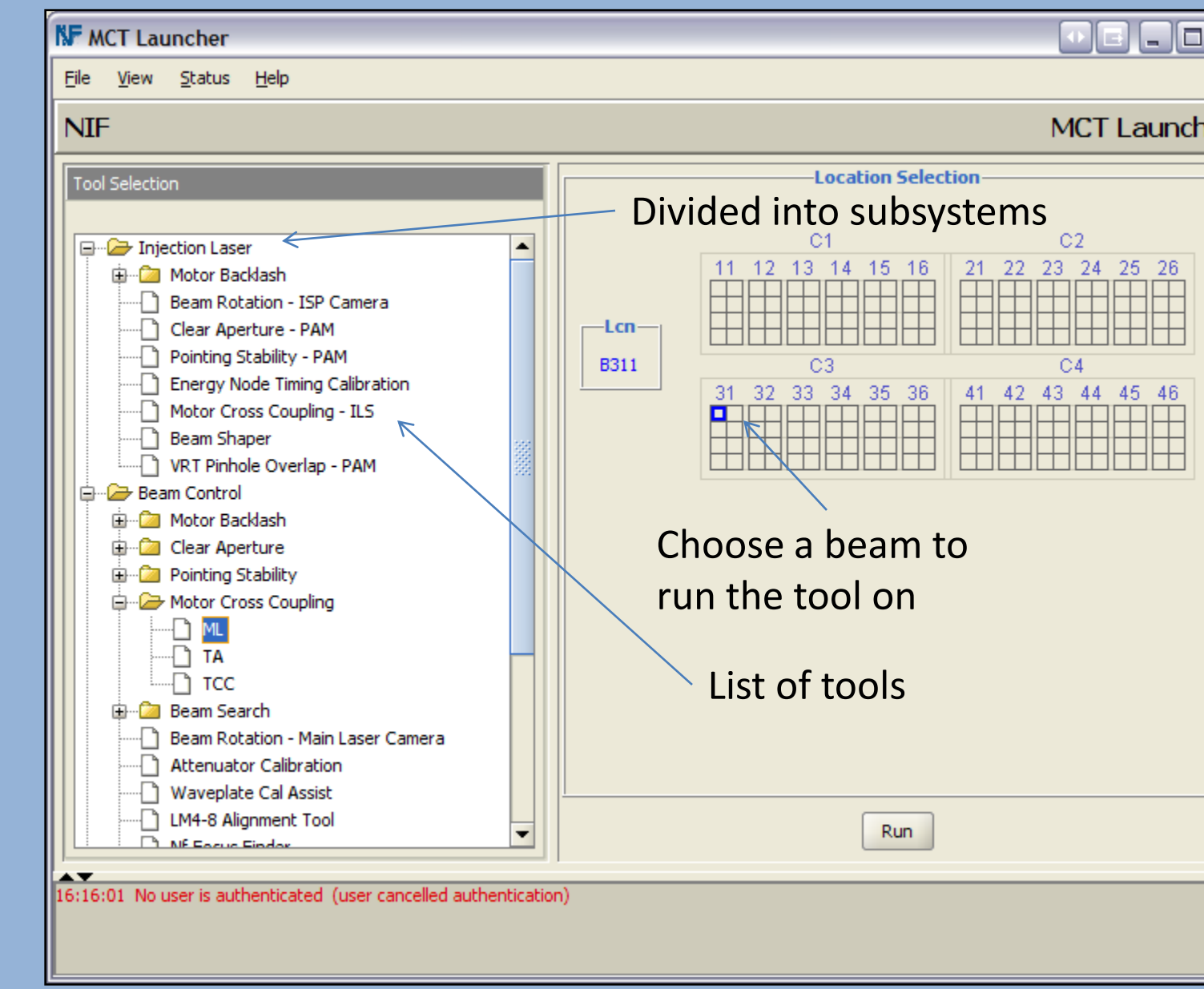
Operator using a MCT



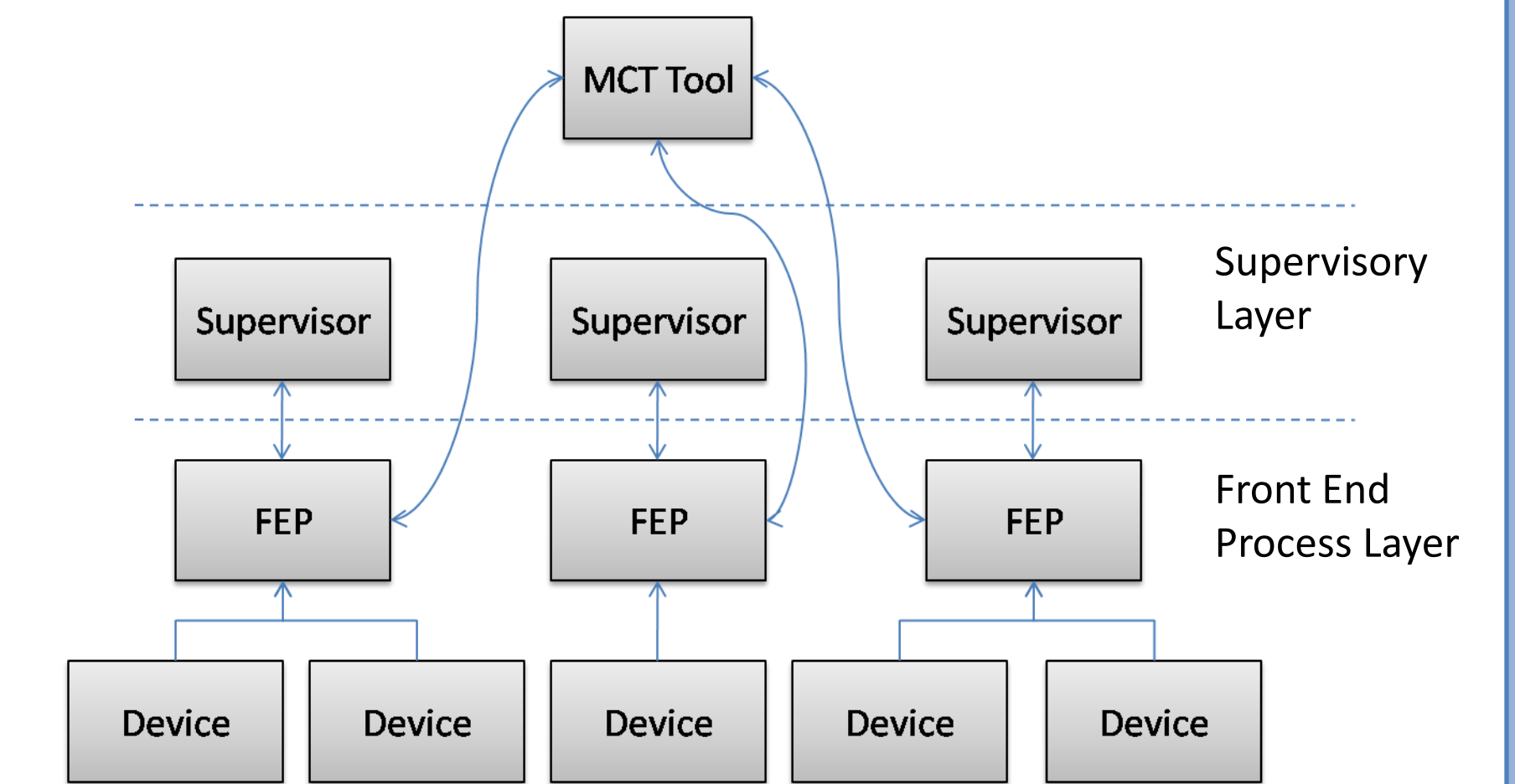
One of 6,200 LRUs in NIF



Central UI to launch tools



Automated tools built on top of ICCS software



### Purpose

- Automate complex procedures
- Capture sequence of actions

### Results

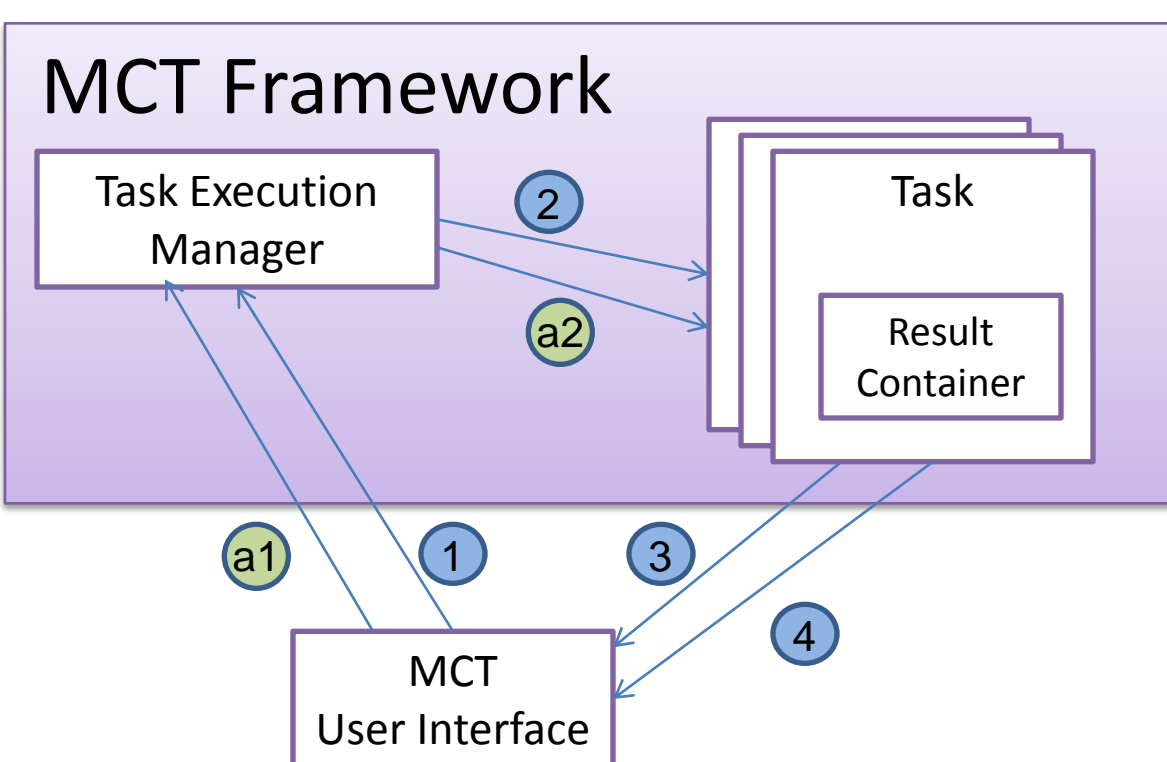
- 25 tools have been delivered to NIF
- Improve efficiency:
  - Minimize operator intervention
  - Reduce delay between actions
  - Run several beams in parallel
- New ways to characterize system

## MCT Framework

### Algorithm Execution

#### Task Execution Management

- Run multiple tasks in parallel
- Stop any running threads
- Thread debugging tools for troubleshooting
- Use Concurrency API in Java 6.0

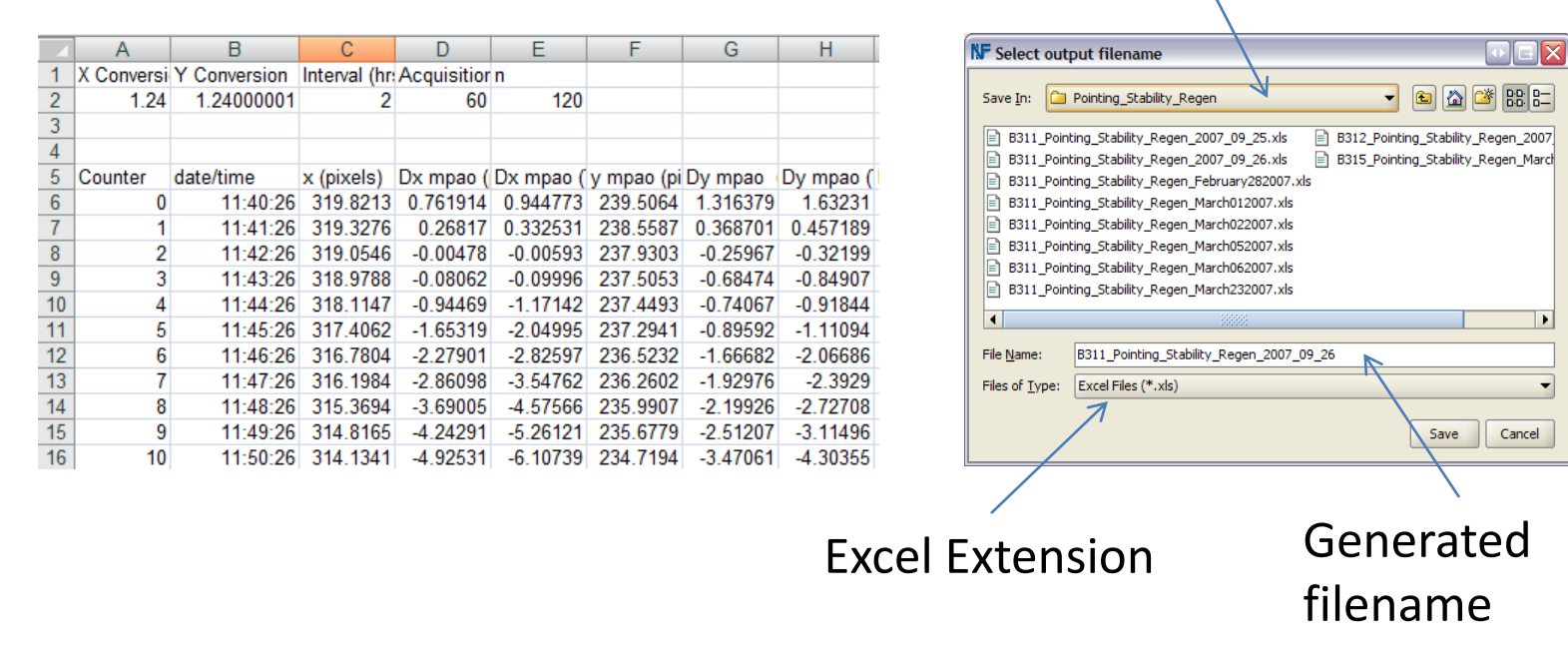


- Request to start execution
  - Create necessary tasks
  - Send progress updates while running
  - Send final results when complete
- a1 At any time, UI may stop algorithm  
a2 Stop request propagated to all tasks

#### Measurement and Results Documentation

- Further offline analysis capability
- Book keep tool's actions
- Central storage location

Regen Pointing Stability Results



Default location for the saved report

Excel Extension

Generated filename

### Storing Configuration Data

#### XML elements for subsystem-specific properties

```

<TOOL NAME="Clear Aperture" SELECTION="BEAM" CLASSNAME="nif.mct.alignment.clear_aperture.ClearApertureDisplay">
  <FILE NAME="CONFIG" VALUE="/clearaperture_config.xml.xsl" />
  <PROP NAME="SEC_CMD_NUM" VALUE="RMN" />
  <PROP NAME="SHOT_ID" VALUE="clear_ap_test" />
  <PROP NAME="ARCHIVE_IMAGES" VALUE="TRUE" />
  <PROP NAME="DEFAULT_TIMEOUT" VALUE="240000" />
  <PROP NAME="SUBSYSTEM" VALUE="BCS" />
  <PROP NAME="TOOL" VALUE="Clear_Aperture_ML" />
</TOOL>
    
```

No restriction on number of files associated with tool

Common data for the tool

#### Additional files for tool-specific needs

Possible Device Names	Default steps for motor move	Data to calculate beam location	Tailor files to suit tool's needs
AC@beam@SY@OM@AS@SH@Y	100 aa_ba_200	TARGET_AREA	FALSE
AC@beam@SY@OM@AS@SH@Y	100 aa_ba_200	TARGET_AREA	FALSE
AC@beam@SY@OM@AS@SH@Y	100 aa_ba_200	TARGET_AREA	FALSE
AC@beam@SY@OM@AS@SH@Y	100 aa_ba_200	TARGET_AREA	FALSE
AC@beam@SY@OM@AS@SH@Y	100 aa_ba_200	TARGET_AREA	FALSE
AC@beam@SY@OM@AS@SH@Y	100 aa_ba_200	TARGET_AREA	FALSE
AC@beam@SY@OM@AS@SH@Y	100 aa_ba_200	TARGET_AREA	FALSE
AC@beam@SY@OM@AS@SH@Y	100 aa_ba_200	TARGET_AREA	FALSE
AC@beam@SY@OM@AS@SH@Y	100 aa_ba_200	TARGET_AREA	FALSE
AC@beam@SY@OM@AS@SH@Y	100 aa_ba_200	TARGET_AREA	FALSE

Add a new device to the tool in less than 5 minutes

### Standardize Common Communication

#### Layer to encapsulate device functions:

- Motors
- Cameras
- Photodiodes
- Automatic Alignment Supervisors

```

//+
// Use the current position
// throws ExecutionException
// throws InterruptedException
public int queryCurrentStepCount() throws ExecutionException, InterruptedException {
    checkInterrupted();
    try {
        int stepCount = motorControl.getPosition();
        updateTaskStatus("Motor" + motorControl.getPosition());
        return stepCount;
    } catch (Exception ex) {
        // The motor's position, ex()
        throw new ExecutionException("Error current position", ex);
    }
}
    
```

Check to see if stopped

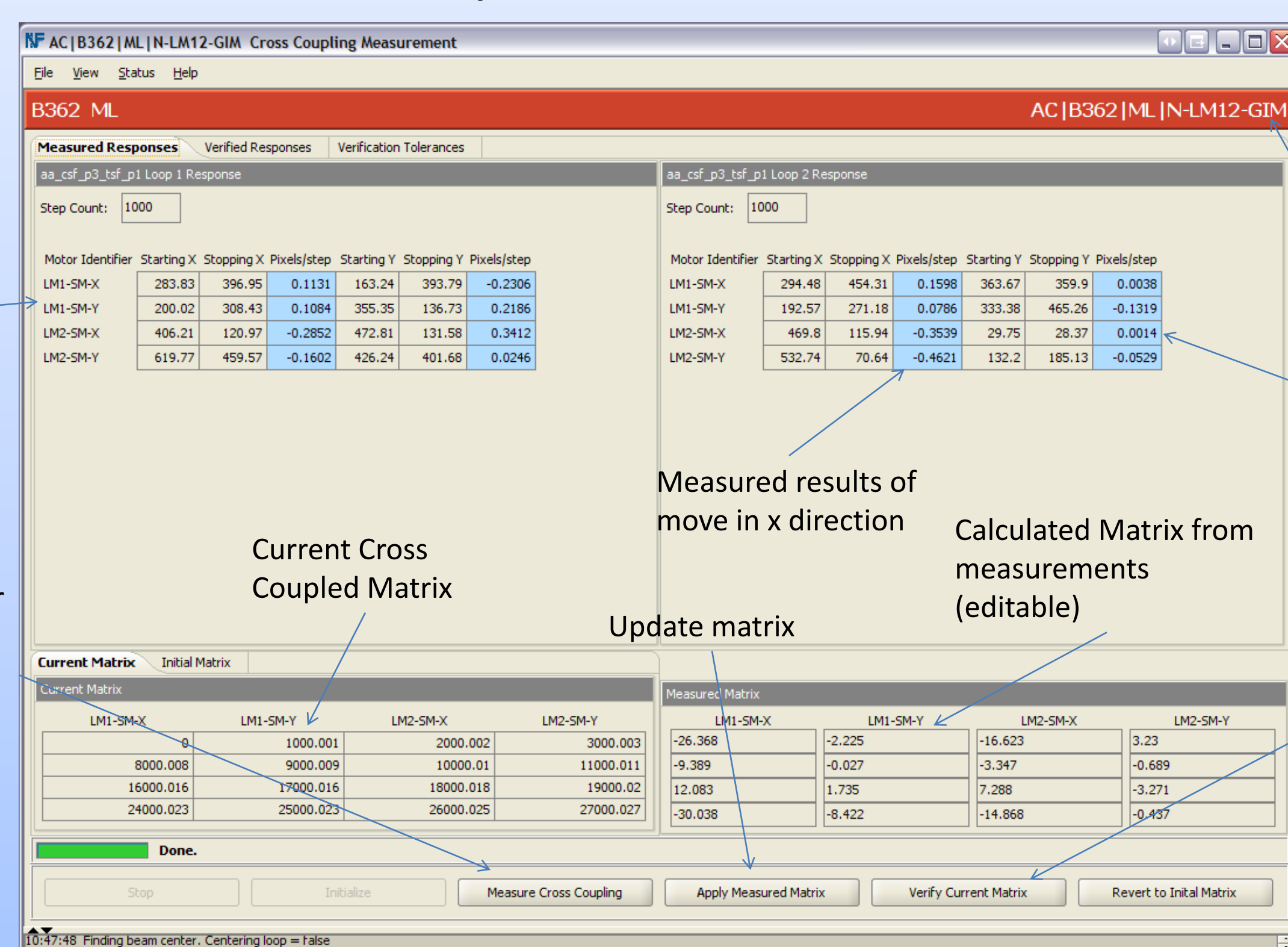
Change to standard Java primitive

Common error message

Standardize status for this method

## MCT Examples

### Cross Coupled Measurement Tool



- Task algorithm:
- Move individual motor
  - Acquire image
  - Find beam center
  - Repeat multiple times for each motor
  - Calculate matrix

Individual motors in gimbal

Current Cross Coupled Matrix

Measured results of move in x direction

Update matrix

Calculated Matrix from measurements (editable)

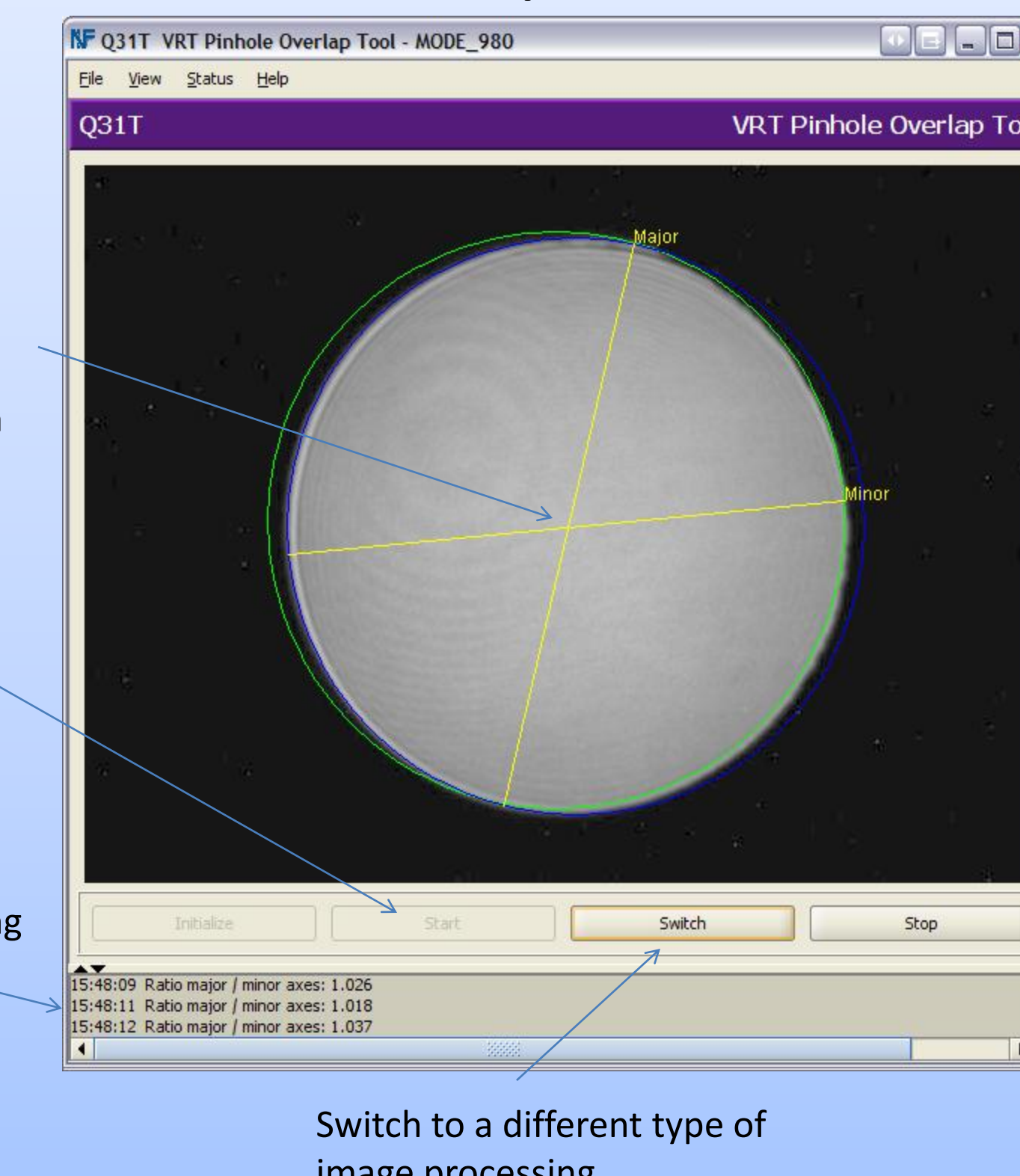
Configuration file determines available gimbals

Measured results of move in y direction

Task algorithm:

- Move gimbal
- Acquire image
- Find beam center
- Calculate difference between expected and actual response

### Pinhole Overlap Tool



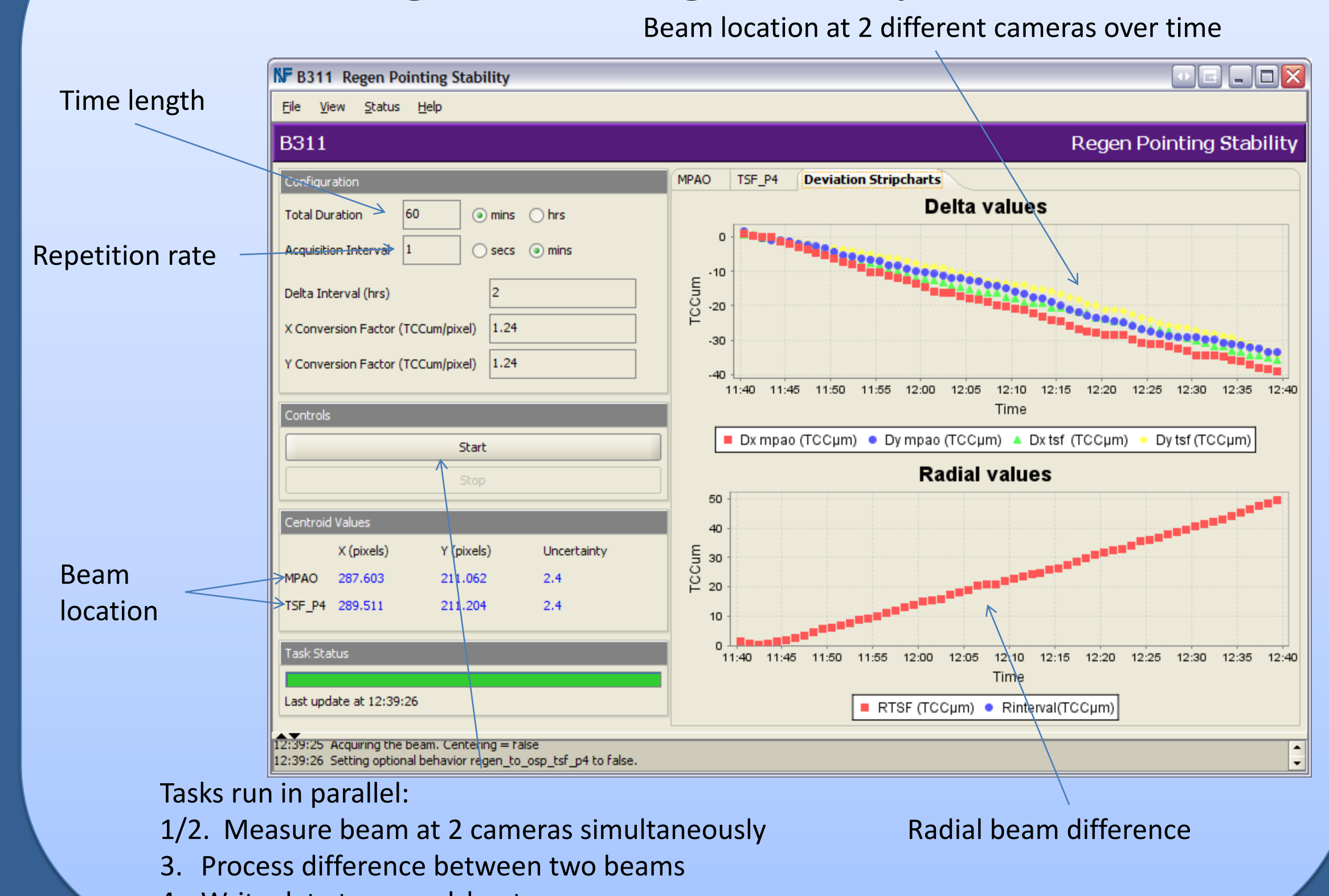
- Custom image analysis
- Find major axis
  - Find minor axis
  - Find pinhole location

Single task captures and processes image continually

Quantify beam clipping

Switch to a different type of image processing

### Regen Pointing Stability Tool



Time length

Repetition rate

Beam location

- Tasks run in parallel:
- 1/2. Measure beam at 2 cameras simultaneously
  3. Process difference between two beams
  4. Write data to spreadsheet

Radial beam difference