MXCuBE status @ ALBA

XALOC Beamline

Jordi Andreu, Control Engineer @ ALBA

MXCuBE meeting, 16th January 2017, Grenoble (ESRF)
Overview
MXCuBE2 @ ALBA

- News
- MXCuBE Integration Strategy
- MXCuBE Current Status
- Next steps
- Project Contribution
News: XAIRA beamline @ ALBA
Microfocus Beamline for Protein Crystallography (XAIRA)

- Phase III beamline.
- PRINCE2 Project management framework.
- MXCuBE (over Sardana) and ISPyB (LIMS).
- Project at the very beginning stage.
- Foreseen operation in 2020.
Integration of MXCuBE 2 (Qt4) @ ALBA
Xaloc Beamline (BL-13)

- Control system: SARDANA + TAURUS + TANGO
- Fully operative beamline...but:
  - Accessibility, current implementation.
  - Missing features (as required by MXCuBE), LIMS, user management.
Integration of MXCuBE 2 @ ALBA

Strategy

- Provide control of individual element through MXCuBE interface: shutters, CATS, diffractometer, ...
- Review/revisit current beamline operation procedures: beam centering, collection methods, ...
- Design & implement missing features (present in MXCuBE): data and users management, ...
- Continuous delivery of MXCuBE features (when possible, standalone applications)

- Take advantage of this situation: improve reliability and beamline robustness.
- Learning from experience (XAIRA design)
Integration of MXCuBE 2 @ ALBA

Current Status

- Qt4 version (master branch).
- Bixente Rey visits.
- MXCuBE interface for all individual beamline elements.
- Two new **Tango DS** required for XALOC:
  - Diffractometer Manager
  - Beamline Supervisor
- **mxCATS** as standalone application.
Integration of MXCuBE 2 @ ALBA

mxCats: standalone application
Integration of MXCuBE 2 @ ALBA

mxCats: expert mode
Integration of MXCuBE 2 @ ALBA
https://github.com/mxcube/mxcube

PR#148: Changes related with SampleChanger.
  ● Allow to derive it for a CatsSimpleBrick (a simplify version of the latter).
  ● Add a "Select" sample on "singleclick"

PR#147: Test environment / Qt4_CatsMaintenance / ALBA specific Bricks.
  ● Create of a module MXCuBE_Test_Environment.py to test HardwareObjects or Bricks
  ● Add a Qt4_CatsMaintenance.py brick (converted from original qt3)
  ● Add a number of ALBA specific Bricks *shutters, Lights, etc…)
  ● Add a convenience QLed widget.
Integration of MXCuBE 2 @ ALBA
https://github.com/mxcube/HardwareObjects

**PR#168:** XMLRPCServer and other ALBA Changes

- Minor changes on XMLRPCServer.py.
- Adding extra optional "shifts" parameter for load, unload commands in CATS.
- Add extra commands in CatsMaint.py.
- Get limits for position from SardanaChannel minval, maxval.
- Add a series of ALBA Specific Hardware Objects.

**PR#171:** LimaVideo and SampleChanger code

- Generic Sample change and Cats90.
- Qt4_LimaVideo (dual library-tango server support).
Integration of MXCuBE 2 @ ALBA

https://github.com/mxcube/HardwareRepository

PR#12 : Sardana.py - finding minval, maxval depending on Taurus version.

- Get limits for SardanaChannel. Make condition depending on Taurus version.
Integration of MXCuBE 2 @ ALBA
https://github.com/mxcube/BlissFramework

PR#14: Small Qt4_widget_colors changes / Qt4_LightControlBrick.

- All colors in Qt4_widget_colors are now instances QColor
- New color_to_hexa() function added in Qt4_widget_colors (to be used in calls of type setStyleSheet())
- New Qt4_LightControlBrick() (converted from Qt3)
Integration of MXCuBE 2 @ ALBA

Next steps

● Implement basic collection methods.
● Design data & user management schemes.
Acknowledgements

The team(s)

**XALOC staff**
- Roeland Boer
- Jordi Juanhuix
- Fernando Gil
- Barbara Machado
- Xavi Carpena

**Controls**
- Guifre Cuni
- Jordi Andreu

**Externals**
- Bixente Rey (TXO)
Thank you for your attention
Integration of MXCuBE 2 (Qt4) @ ALBA

Evolution of MXCuBE implementation @ ALBA

Bessy (June 2015)

ALBA (December 2015)

DESY (June 2016)

ESRF (February 2017)
Integration of MXCuBE 2 @ ALBA

Current status

- Under development for XALOC Beamline (BL13) @ ALBA.
- Basic operations available (centering, autofocus, etc…)
- Early stage on the Qt4 implementation (*Qt3-*→*Qt4 overhead*).
- *Limited access* for development due to beamline in fully operation.

**we were:**

<table>
<thead>
<tr>
<th>Developing on branch 2.1</th>
<th>Qt3 Graphical Interface</th>
<th>EDNA @ workstation</th>
</tr>
</thead>
</table>

**we are:**

<table>
<thead>
<tr>
<th>Developing on branch 2.2 (stable)</th>
<th>Qt4 Graphical Interface</th>
<th>EDNA @ cluster</th>
</tr>
</thead>
</table>
Integration of MXCuBE 2 @ ALBA

New implementations

HardwareObjects: PR #168 (XMLRPCServer and other ALBA Changes (New objects, Cats evolution, SardanaMotor)

- Adding extra optional "shifts" parameter for some of the CATS commands allow to specify "shifts" in load, unload commands in CATS.
- Add extra commands in CatsMaint Add some more commands to CatsMaint.py.
- Update SardanaMotor.py. Get limits for position from SardanaChannel minval, maxval.
- ALBA Hardware Objects. Add a series of ALBA Specific Hardware Objects.
Integration of MXCuBE 2 @ ALBA

XALOC: Current status
Qt4 Graphical Interface

General: Camera in Qt4 (PR # 74 to branch 2.2)

- Modifications to the `Qt4_LimaVideo.py` HwObj to provide support for Basler cameras *via LImA* library.
- Supported pixel types:
  - *YUV 422 packed* (Color)
  - *Y8* (Black and White)
- The implementation is *QUB* independent.
- Depends on *OpenCV* library python interface.
- Easily extensible to other pixel formats.

**Configuration:**

```
<device class="Qt4_LimaVideo">  
  <type>basler</type>  
  <address>84.89.227.72</address>  
  <encoding>yuv422p</encoding>  
  <gain>0.3</gain>  
  <exposure>0.01</exposure>  
  <interval>30</interval>  
</device>
```
Qt4 Graphical Interface

ALBA Specific: Xaloc Hardware Objects (PR # 73 to branch 2.2)

- The parent class of (ALBA specific) Xaloc\texttt{Minidiff} HwObj has been changed from \texttt{MiniDiff} to \texttt{GenericDiffractometer} class.
- Hutch menu brick works fine: Centering routines, focus (sardana macro), snapshot, etc...
- \textbf{BUT} we still using the old centering routines module.

- \textbf{Overhead:} We had the need to adapt some HardwareObjects which were already running for Qt3 version: MachineInfo, SafetyShutter, Beamstop, etc...
EDNA plugins
Strategy & Fast autoproc plugins

- **EDNA Strategy: EDPluginControlInterfaceToMXCuBEv1_3**
  - Already in production (Tango EDNA server).
  - Running on workstation (8 cores) Intel(R) Xeon(R) CPU E31275 @ 3.40GHz.

- **EDNA Fast Autoproc plugin: EDPluginControlAutoprocV1_0**
  - Testing @ Cluster (direct batch queue).
  - Running on a HPC node (16 cores) Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz
EDNA performance
Strategy & Fast autoproc plugins

- Lysozyme (1800 images set)

Cluster geometry for XDS
http://xds.mpimf-heidelberg.mpg.de/html_doc/xds_parameters.html

<table>
<thead>
<tr>
<th>config</th>
<th># procs</th>
<th># jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>32</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>D</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>32</td>
<td>1</td>
</tr>
</tbody>
</table>

JOBS: independent subprocesses
PROCS: parallel OpenMP
MXCuBE 2 @ ALBA

Based on SARDANA & TAURUS Control System
Integration of MXCuBE 2 @ ALBA

Over Sardana Control layer

Graphical User Interface
- Bliss Framework
  - python 2.x and Qt4
- MXCuBE Bricks
- Specific Bricks for ALBA/Xaloc

Control Layer
- Hardware Repository
  - python class + xml configuration file
- MXCuBE Hardware Objects
- Specific Hardware Objects for ALBA/Xaloc
- Taurus / Sardana support
- XML configuration files
Integration of MXCuBE 2 @ ALBA
Sardana Support for Hardware Repository (V. Rey)

*Implemented by V. Rey in CommandContainer.py and Sardana.py files from Hardware Repository (ALBA branch @ github)
Integration of MXCuBE 2 @ ALBA
Sardana Support for Hardware Repository

Hardware Object configuration (Sardana Layer):

```xml
<object class="SardanaXfeSpectrum">
  <doornname>door/mxcube/1</doornname>
  <command type="sardana" name="configure">senv ActiveMntGrp mg_fludet</command>
  <command type="sardana" name="acquire">ct %s</command>
  <channel type="sardana" taurusname="fluodet_timer" name="spectro">spectrum</channel>
</object>
```
Integration of MXCuBE 2 @ ALBA
Sardana Support for Hardware Repository

Sardana Motor:

Position and State (channels) and Stop (command) are defined by default...

```xml
<device class="SardanaMotor">
    <username>Omega Z</username>
    <taurusname>omegaz</taurusname>
</device>
```

...even they can specified in the configuration file:

```xml
<device class="SardanaMotor">
    <username>Omega Z</username>
    <taurusname>omegaz</taurusname>
    <channel type="sardana" polling="events" name="position">Position</channel>
    <channel type="sardana" polling="events" name="state">State</channel>
    <command type="sardana" name="stop">Stop</command>
</device>
```