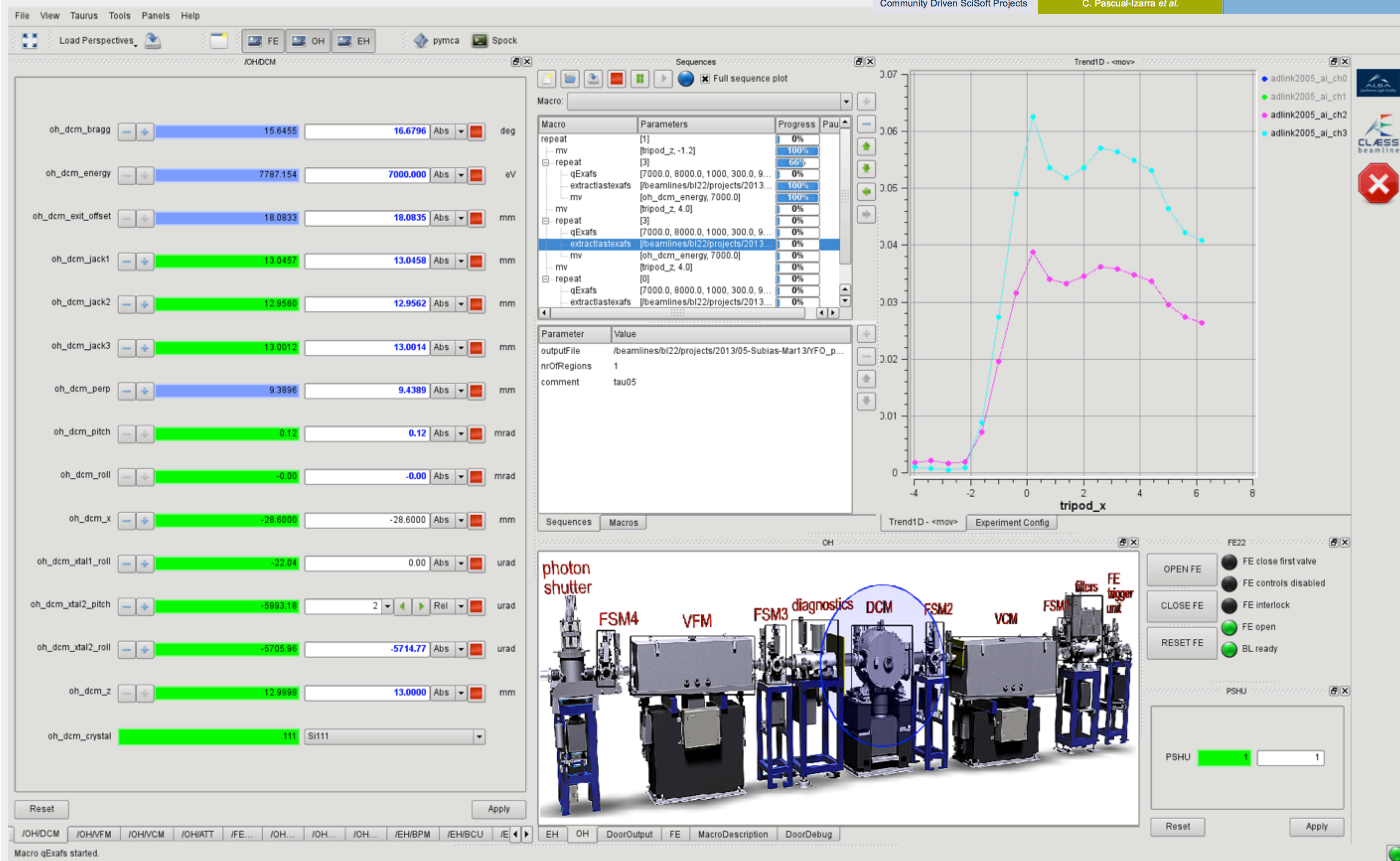
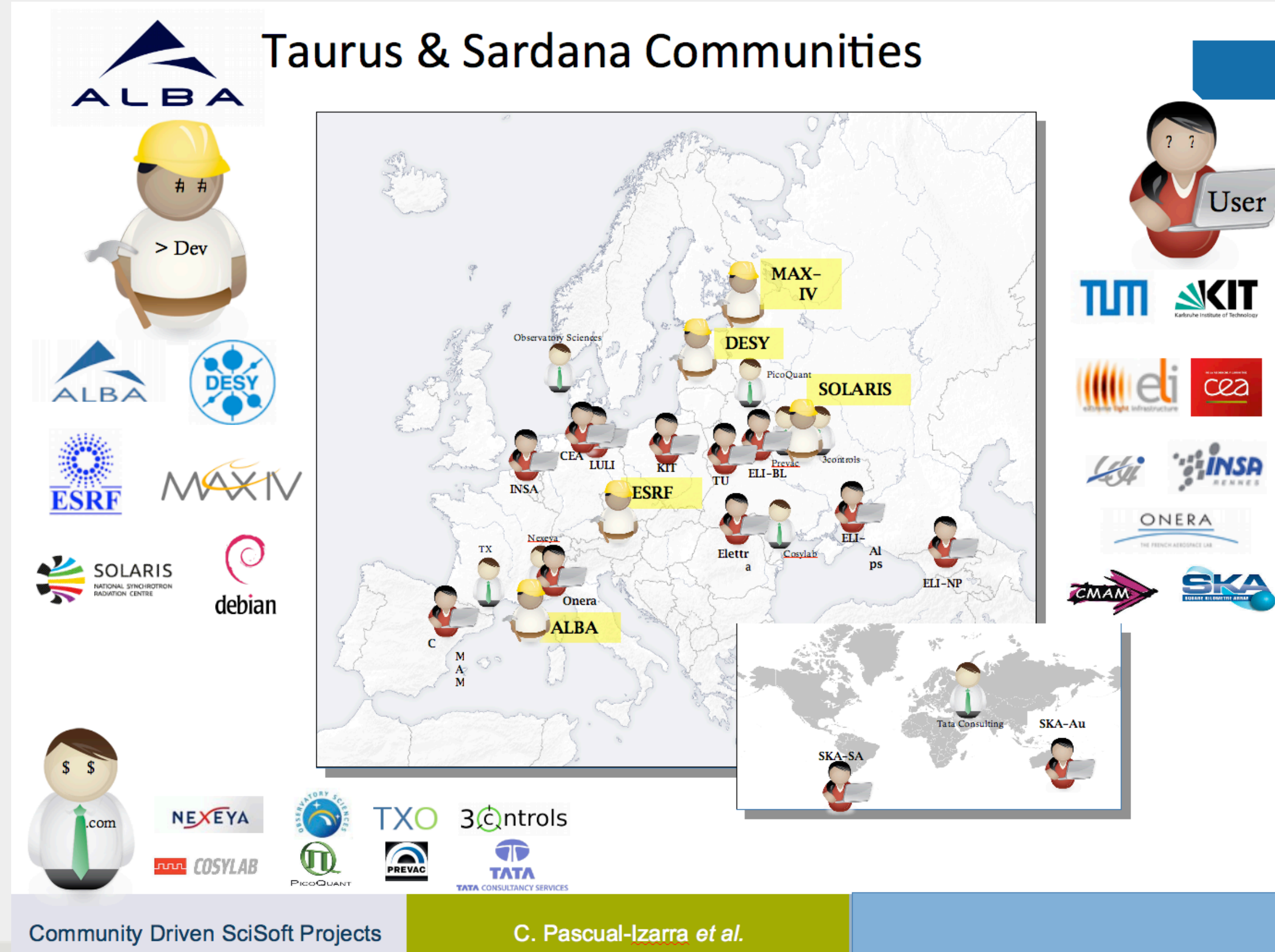




BUILDING S.C.A.D.A. SYSTEMS IN SCIENTIFIC INSTALLATIONS WITH SARDANA AND TAURUS

D. Fernandez-Carreiras, J. Andreu, F. Becheri, S. Blanch-Torné, M. Broseta, G. Cuni, C. M. Falcon-Torres, R. Homs-Puron, G. Jover-Mañas, J. Klorá [on leave], J. Moldes, C. Pascual-Izarra, S. Pusó-Gallart, Z. Reszela, V. Rey-Bakaikoa [on leave], D. Roldan, M. Rosanes-Siscart, A. Rubio, S. Rubio-Manrique, J. Villanueva (ALBA-CELLS, Cerdanyola del Vallès), T. Kracht, M. T. Nunez Pardo de Vera (DESY, Hamburg), T. Coutinho, A. Homs, . Taurel (ESRF, Grenoble), V. Hardion, A. Milan, D. P. Spruce (MAX IV, Lund), P. Goryl, L. Dudek, L. Zytniak (Solaris, Krakow) Lon behalf of the Sardana and Taurus communities.

Sardana and Taurus form a python software suite for Supervision, Control and Data Acquisition (SCADA) optimized for scientific installations. Sardana and Taurus are open source and deliver a substantial reduction in both time and cost associated to the design, development and support of control and data acquisition systems. The project was initially developed at ALBA and later evolved to an international collaboration driven by a community of users and developers from ALBA, DESY, MAXIV and Solaris as well as other institutes and private companies.



Main human interface. Beamline BL22-CLAESS at Alba-Synchrotron

```
class set_lm(Macro):
    """Sets the dial limits on the specified motor"""
    param_def = [
        ['motor', Type.Motor, None, 'Motor name'],
        ['low', Type.Float, None, 'lower limit'],
        ['high', Type.Float, None, 'upper limit']
    ]

    def run(self, motor, low, high):
        name = motor.getName()
        self.debug("Setting dial limits for %s" % name)
        motor.getDialPositionObj().setLimits(low, high)
        self.output("%s limits set to %.4f %.4f (dial units)" % (name, low, high))

(a)
LAB-01-D01 [1]: ascan
Syntax:
ascan <motor> <start_pos> <final_pos> <nr_interv> <integ_time>

Do an absolute scan of the specified motor.
ascan scans one motor, as specified by motor. The motor starts at the
position given by start_pos and ends at the position given by final_pos.
The step size is (start_pos-final_pos)/nr_interv. The number of data points collected
will be nr_interv+1. Count time is given by time which if positive,
specifies seconds and if negative, specifies monitor counts.

Parameters:
motor : (Motor) Motor to move
start_pos : (Float) Scan start position
final_pos : (Float) Scan final position
nr_interv : (Integer) Number of scan intervals
integ_time : (Float) Integration time

(b)
```

- The macro execution environment, multiclient and and interactive is combined with the graphical interfaces the heart of the user experience.
- The communication layer is Tango, although Taurus interfaces can connect with various schemas. The data is handled and presented in various formats, depending on the needs and on the appropriate format for the nature of data.
- Due to the open source nature of the project, the Sardana development is likely to gain considerable momentum. The latest release, published in of 2017, improves its performance, as well as the experiment configuration, plotting and access to data files. It also natively integrates continuous scans. Looking ahead to the upcoming versions, the efforts of the community will be focused among others in the configuration tools, full integration of 2D detectors, and continuous scans frameworks with complex movements and trajectories.

