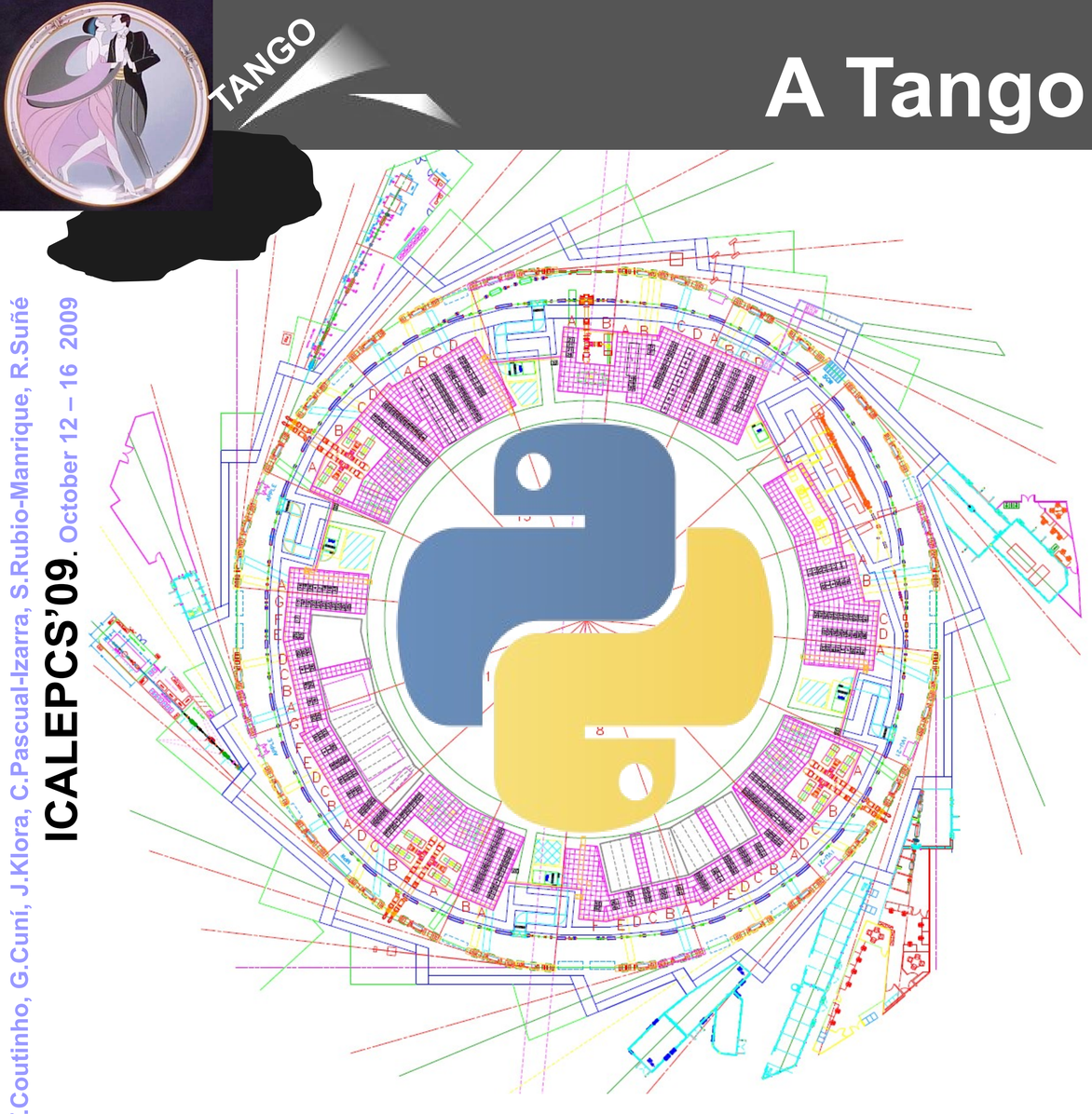
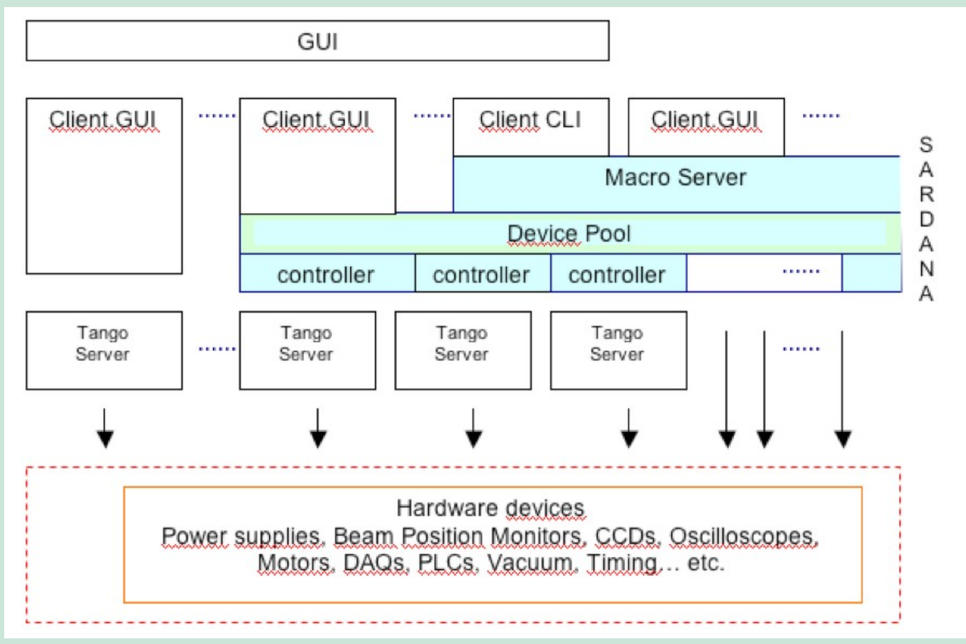
A Tango Based Control System in Python

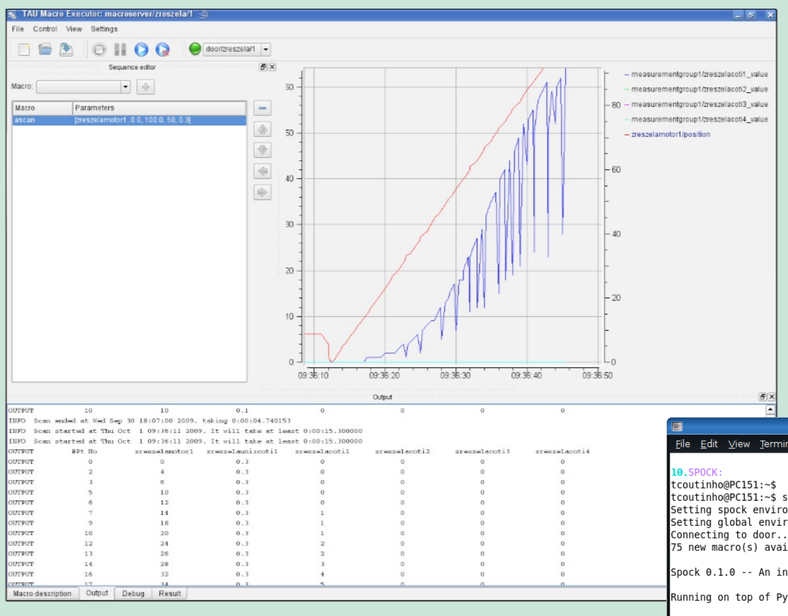
D. Fernández-Carreiras, F.Becheri, S.Blanch, T.Coutinho, G.Cuní, J.Klora, C.Pascual-Izarra, S.Rubio-Manrique, R.Suñé

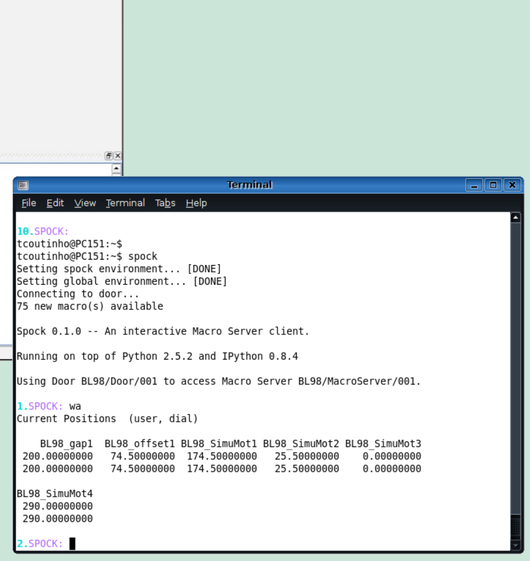
ICALEPCS'09. October 12 – 16 2009



The **Tango** collaboration counts now five members: chronologically, the ESRF (where Tango was initially developed), Soleil (The first synchrotron using Tango as the unique controls middleware), Elettra, Alba and DESY. The Python support for Tango (**PytTango**) has been developed at Alba by E. Taurel (now at the ESRF) and T. Coutinho. PyTango is now extensively used at Alba.

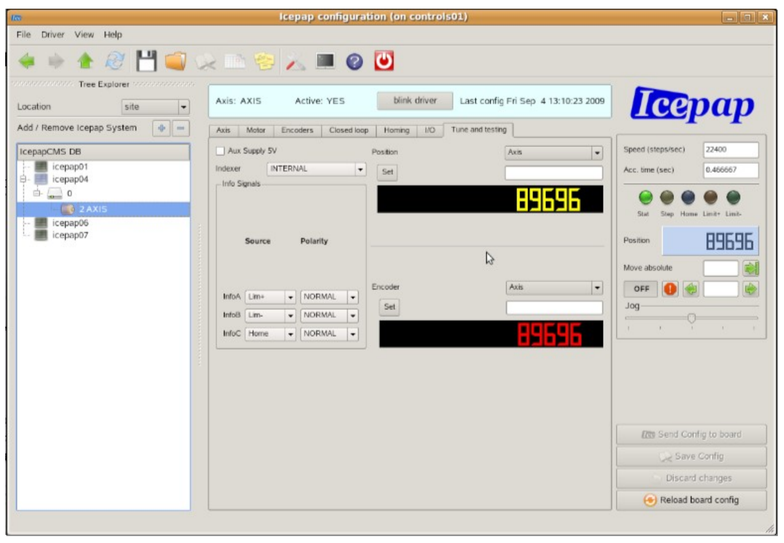


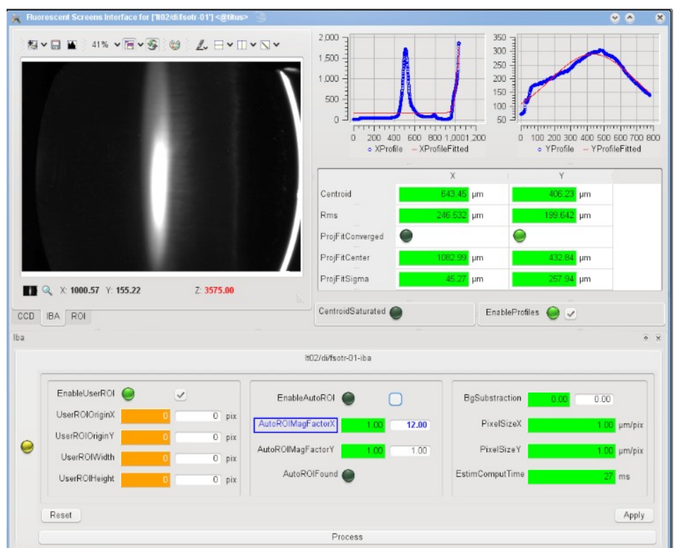


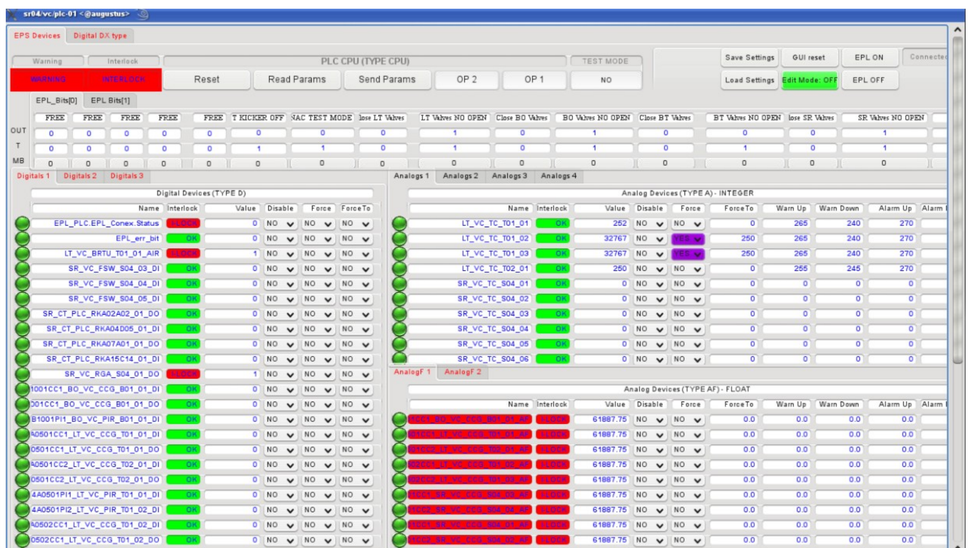


Spock is a IPython based **CLI** for **Sardana**. It communicates with the macroserver through one of its doors. Because it is IPython based, all CLI niceties (like word completion on Tab, command history on Up, Down arrows) are automatically available which makes it very easy for spock to mimic SPECs UI. The user executes 'magic' commands (like scans, alignment procedures) in the command line which spock translates to execution of macros inside the macroserver.

The **MacroExecutor** is a widget offering a user-friendly interface to the macro repository. Standard and user defined macros are selected from a graphical user interface, prompting for input arguments and offering results and plots







"Stand-Alone" applications. IcepapCMS.
Icepap *is the standard motor controller at Alba. The hardware has been developed at the* ***ESRF****, and the* ***configuration software*** *at* ***Alba (Icepapcms).*** *Many parameters, some of them critical, must be configured before using a motor, like current, encoders, limits, etc. It manages a database with historical configurations; supports motor catalogs and has a tool for online tests. Icepapcms is built with the same standard tools.*

TAU. GUIs built on QT. On the right the Fluorescence Screen GUI. On the left the PLC-Equipment protection Expert GUI.
It implements a model view controller pattern. The models are provided by the TauCore layer, which is independent of the widgets. Normally GUIs are generated by the Qt Designer, but not exclusively. A library of Tau Widgets is available also from the designer. Written mostly in Python although, some of them in C++ and integrated with sip, these set of widgets constitute the basic bricks for creating graphical applications or new widgets. The GUI for fluorescence screens shows some widgets. QUB (developed at the ESRF shows the image), 2 TauPlot widgets show profiles of the image (1D). TauForms show/edit parameters. The Expert PLC-Equipment Protection GUI is automatically generated from the Cabling database.

CONCLUSION

Alba has focused the software developments on Python. Most device servers and all graphical user *interfaces are written in Python. It has proven to be suitable for GUIs (combined with Qt), and for most device servers. It is also very convenient for writing simulations and dynamic attribute expressions. Moreover it is an excellent choice for data analysis. Only the cases where very fast data acquisition, storage or processing is needed complementary solutions are proposed.*

Many developers are working on this project. We would like to thank all of them. E. Taurel is the main Tango Core developer and the first author of the Sardana Framework and the Tango binding for Python. Besides the ones named in the Author Tab, there are many other people working in the project: L. Krause (Power Supplies, Linac), Z. Reszela (Tau widgets, beamlines), J. Moldes (Libera BPMs, Timing, beamlines), A. Milán (RF, beamlines), M. Niegowski (Radiation Monitors, PLC GUIs), We would also like to thank other contributors from different institutions, in particular A. Homs, V. Rey, S. Petitdemange, G. Berruyer, L. Claustre, M. Guijarro and the whole Bliss group at the ESRF, for their great collaboration. Not forgetting T. Nuñez and T. Kracht at DESY, and all our Partners in the Tango Community.