A BOTTOM-UP APPROACH TO AUTOMATICALLY CONFIGURED TANGO CONTROL SYSTEMS

S. Rubio-Manrique, D. Beltran*, I. Costa, D. Fernandez-Carreiras, J.V. Gigante, J. Klora, O. Matilla, R. Ranz*, J. Ribas*, O. Sanchez, CELLS-ALBA Synchrotron, Cerdanyola del Vallés, Spain

Abstract

Alba maintains a central repository, so called "Cabling and Controls database" (CCDB), which keeps the inventory of equipment, cables, connections and their configuration and technical specifications. The valuable information kept in this MySQL database enables some tools to automatically create and configure Tango devices and other software components of the control systems of Accelerators, beamlines and laboratories. This paper describes the process involved in this automatic setup.

INTRODUCTION

ALBA is the first Synchrotron Light Source built in Spain[1]. Its 3 GeV Storage Ring has been commissioned during 2011, receiving in 2012 the first users for the 7 beam-lines. Most of ALBA control system [2] have been developed on top of Tango Control System [3][4].

An amount of 5531 devices are controlled up-to-now in ALBA accelerators (linac, booster and storage ring) using 150 control Linux PCs. Those devices are accessed from the control system using several hardware interfaces (ethernet, fiber optics, serial lines, GPIB, PLC inputs, …) that had to be configured in the Tango database.

To populate the tango database, information related to hardware configuration and connections has been imported from ALBA's Cabling and Controls Database (CCDB) [5]. The cabling database keeps information of all hardware connections and the network configuration of all ethernet equipments. We used this information already in CCDB to automatize the creation and configuration of several control subsystems, like vacuum, front-ends, motor controllers and EPS [6]; speeding-up the process and minimizing the errors.

THE CABLING AND CONTROLS DATABASE

The Cabling and Controls database was developed in-house by ALBA management and information software section (MIS) to be used as the main knowledge support for design, control, tendering, progress follow up and a search tool providing multiple views of the system.

This MySQL database has been used since 2007 to keep a knowledge base of all the equipments and devices used at ALBA accelerators and beam-lines. The Cabling database keeps all the information about equipment types, cable configurations, connections between devices distribution of hardware in racks, routing of cables between equipment and network configuration. It also provides fast access to all available documentation for each type of equipment. It actually contains in 346 racks with 6326 equipments of different 638 equipment types. These equipments are connected using 17623 cables of 292 different cable types with a total length of 167,43 Km.

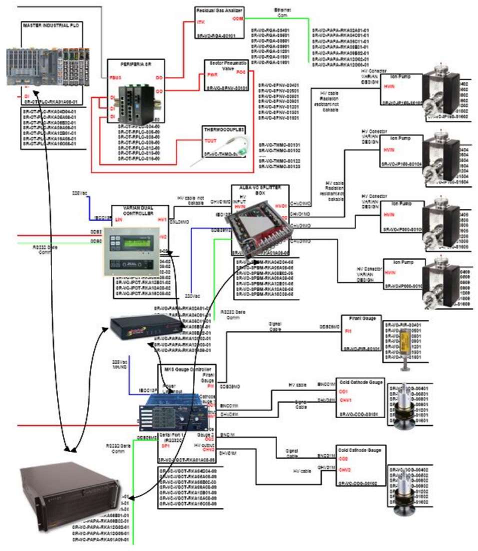


Figure 1: Diagrams of every subsystem have been produced prior to its introduction in our cabling database.

Introducing Cables in the Cabling Database

Filling this database required and intensive effort by a group of engineers (one dedicated to each subsystem). They introduced most of the information in the cabling database during 2008, although installation continues and the database is updated almost daily. All the documentation used to populate the database has been kept attached to each equipment and rack; so it is possible to download original diagrams (Fig. 1) and equipment manuals from the CCDB web client (Fig. 2).

The maintenance of the database is managed by a group of electronic engineers. Modification or insertion of new cables in the database is done using spreadsheet files, that contain source and destination equipment-terminal pairs and cable types. Every file is cross-checked prior to insertion to avoid errors and collision with previous data.

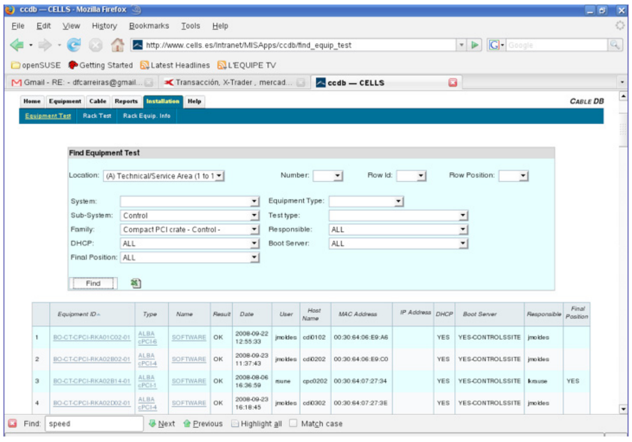
The Cabling Database API

The common method for visualization and modification of information in the cabling database is the CCDB web client. This tool browses all the information in the database and provides many methods for viewing and exporting the data to text files. The first integration with

*On leave

the control system was based on reading these exported files, but soon a better interaction was needed to guarantee a system up-to-date.

Now the CCDB database becomes available to the control system using the CCDB python API. This python module provides methods to search for equipments and get lists of connections, names and network information. Some translation features have been added to the API to store links between every Control and Hardware object, this links will allow every single script to update or cross-check the properties of a Control object from the CCDB.


ccdb — CELLS - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.cells.es/Intranet/MISApps/ccdb/find_equip_test
openSUSE Getting Started Latest Headlines L'EQUIPE TV
Gmail - RE: - dfcarreiras@gmail...
Transacción, X-Trader , mercad...
ccdb — CELLS
Home Equipment Cable Reports Installation Help
CABLE DB
Equipment Test Rack Test Rack Equip. Info
Find
Find: speed
Next Previous Highlight all Match case

Find Equipment Test

Location: (A) Technical/Service Area (1 to 1
Number:
Row Id:
Row Position:
System:
Equipment Type:
Sub-System: Control
Test type:
Family: Compact PCI crate - Control -
Responsible: ALL
DHCP: ALL
Boot Server: ALL
Final Position: ALL

	Equipment ID	Type	Name	Result	Date	User	Host Name	MAC Address	IP Address	DHCP	Boot Server	Responsible	Final Position
1	BO-CT-CPCI-RKA01C02-01	ALBA cPCI-6	SOFTWARE	OK	2008-09-22 12:55:33	jmoldes	cdi0102	00:30:64:06:E9:A6		YES	YES-CONTROLSSITE	jmoldes	
2	BO-CT-CPCI-RKA02B02-01	ALBA cPCI-4	SOFTWARE	OK	2008-09-23 11:37:43	jmoldes	cdi0202	00:30:64:06:E9:C0		YES	YES-CONTROLSSITE	jmoldes	
3	BO-CT-CPCI-RKA02B14-01	ALBA cPCI-1	SOFTWARE	OK	2008-08-06 16:36:59	rsune	cpc0202	00:30:64:07:27:34		YES	YES-CONTROLSSITE	kmuse	YES
4	BO-CT-CPCI-RKA02D02-01	ALBA cPCI-4	SOFTWARE	OK	2008-09-23 16:18:45	jmoldes	cdi0302	00:30:64:07:27:3E		YES	YES-CONTROLSSITE	jmoldes	

Figure 2: The cabling database web client.

GENERATING A CONTROL SYSTEM

Creating Vacuum Controls Devices

The accelerators vacuum control, with 1496 devices, is the largest control system in our Tango database. Creation of vacuum devices have been done automatically for all subsystems, using the cabling database python API to extract all needed information and all connections between equipments.

For each vacuum device (gauge, pump or valve) that belongs to our accelerators we obtain:

- Port and controller reading the device.
- Serial line and Controls computer connected to the controller
- PLC managing the pressure interlock
- Valves controlled by this interlock.

A Tango database entry is generated for each device and every configuration value. We initially used the same text format as Jive (the default Tango configuration tool), but finally we moved to comma-separated values files. A column based format allowed us more flexibility to add some Tango information like device host and run-level. These methods for massive insertion and modification of Tango database have been added to functional-Tango [7] python module, available from Tango repositories.

The architecture of hardware and software objects may differ, not having a one-to-one conversion between CCDB and Tango. The translation features of the API allow to link a different Tango device with each connector of a patch panel (e.g. serial lines) or link a single complex devices with all the software controllers of its parts (e.g. a single mirror translates into a group of gauges, pumps, motors and thermocouples). This translation allows to rebuild the whole control architecture from the cabling.

Distributing Tango Devices in Control Hosts

The Tango devices (control software processes) declared in our database are distributed in 150 PCs in our racks area. Although the control of most equipments is distributed by location (e.g. front-end devices are controlled from dedicated front-end racks) some subsystems have no dedicated racks and use spare ports of other devices, optimizing the number of devices and length of cabling connections.

Furthermore, every Tango process is assigned to a particular host from the CCDB, following the connection between the gauge in the tunnel to a controller in the rack and then the serial line from the controller to the patch panel of that particular industrial PC where the serial line controller will be configured.

Configuring Devices

There are many parameters that are configured in the Tango control database using the information updated from the CCDB. The most common is the number of ports to be used in those devices that have multiple configurations (serial lines, ADC´s, ion pump splitters), the device configuration will get this value from the CCDB to configure which channels will be used and which will be ignored when connecting to the hardware. Also dynamic attributes will be created for used channels only.

Another example are the sizes of ion pump, this value will be used by the control system when converting the current readings to an estimated pressure value. In some cases the sizes of ion pumps change. For example, when vacuum sections are replaced by insertion devices. These changes must be updated in the Tango database and the information is queried from CCDB.

Information relative to networking (IP, masks, host-names, routing) helped to control or update existing hardware that is interfaced with the control system using TCP/IP. This allows to check and restore the status of these devices (splitters, Icepaps, DAQs,) after a powercut.

Keeping the Databases up to date.

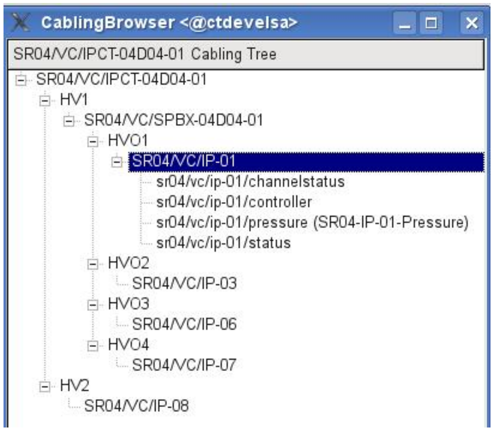
During the last year several modifications have been done in our accelerators. Diagnostic elements have been moved and several vacuum chambers have been replaced by new insertion devices. These changes required to modify racks and cables and introduce the new equipment and connections in the graphical applications.

In case that any change of the routing is done, the technicians will update the information in the CCDB and the control system can detect the change and advice to update it. New CSV files, containing device names and all its property values, are extracted from either CCDB or

Tango databases and are compared to detect missing information, errors or not applied changes.

INTERACTION WITH GRAPHICAL INTERFACES

The information available in the cabling database is used to extend the displays of the control system and allocate equipments in generic browsers. The cabling API enables applications to sort and search by location of controllers or by location of the gauges connected to them (racks and equipments in tunnel don't follow the same name). It allows to group equipments affected by same logics and also make them findable in the racks area when searching for problems.

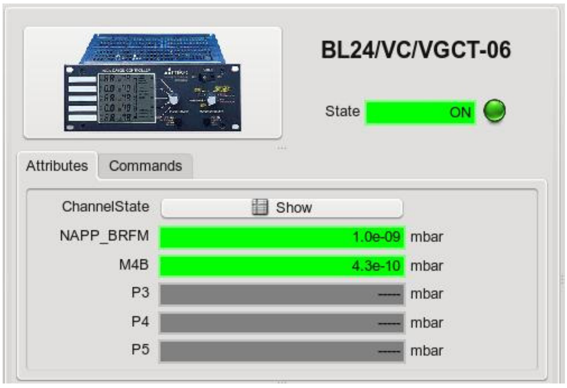


```
CablingBrowser <@ctdevelsa>
SR04/VC/IPCT-04D04-01 Cabling Tree
⊟ SR04/VC/IPCT-04D04-01
  ⊟ HV1
    ⊟ SR04/VC/SPBX-04D04-01
      ⊟ HVO1
        ⊟ SR04/VC/IP-01
            sr04/vc/ip-01/channelstatus
            sr04/vc/ip-01/controller
            sr04/vc/ip-01/pressure (SR04-IP-01-Pressure)
            sr04/vc/ip-01/status
      ⊟ HVO2
          SR04/VC/IP-03
      ⊟ HVO3
          SR04/VC/IP-06
      ⊟ HVO4
          SR04/VC/IP-07
  ⊟ HV2
      SR04/VC/IP-08
```

Figure 3: Tango devices can be shown in trees using the physical connections between equipments.

This extra information is added to the GUI's in two ways:

- In the Tango Database, adding cabling names as aliases of devices or labels of attributes (Fig. 4).
- In cabling-enabled widgets, browsing widgets (Fig. 3 and Fig. 5) that translate Tango names into CCDB to group the devices or attributes using their location or connections.



BL24/VC/VGCT-06

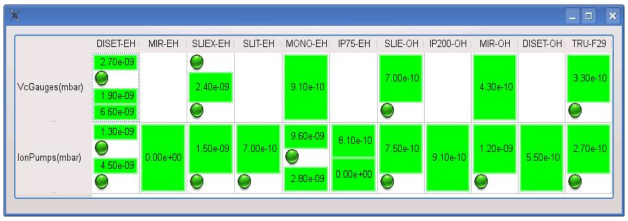
State ON ●

Attributes | Commands

ChannelState	Show	
NAPP_BRFM	1.0e-09	mbar
M4B	4.3e-10	mbar
P3	-----	mbar
P4	-----	mbar
P5	-----	mbar

Figure 4: Names of vacuum gauge ports are labelled with the tag of the device connected to them.

When ports of a same controller are used to read different equipments (e.g. an in-vacuum mono-chromator and a mirror separated by valves) the grid widget (Fig. 5) scans the controllers in the vacuum rack and then it obtains the experimental instrument connected to each port. This information is used to reorganize values in panels and assign labels to pressure values.



	DISET-EH	MIR-EH	SLIEX-EH	SLIT-EH	MONO-EH	IP75-EH	SLIE-OH	IP200-OH	MIR-OH	DISET-OH	TRU-F29
VcGauges(mbar)	2.70e-09 ● 1.90e-09 6.60e-09		● 2.40e-09 ●		9.10e-10		7.00e-10 ●		4.30e-10		3.30e-10 ●
IonPumps(mbar)	1.30e-09 ● 4.50e-09 ●	0.00e+00	1.50e-09 ●	7.00e-10 ●	9.60e-09 ● 2.80e-09	8.10e-10 0.00e+00	7.50e-10 ●	9.10e-10	1.20e-09 ●	5.50e-10	2.70e-10 ●

Figure 5: Labels from cabling are used to group in grids the experimental equipment pressures.

CONCLUSIONS

Automated creation and configuration of big Control systems is a recurrent topic due to its applications in big facilities, reducing the amount of work and human errors when configuring thousands of devices. Acquiring and maintaining a cabling database is hard to do in a running facility, but it is a unique opportunity for new machines as this kind of database is a valuable tool at all levels.

An important application of automated configuration is the fine-tuning of the whole control system, modifying configuration parameters in many devices at the same time (new time-outs in all serial lines, different polling period in different groups of devices, update all host names at once). We used it during commissioning to optimize our control system performance, having also a fast way to do configuration roll-back.

We have now an automated way to load new values to all properties: reducing the time needed to deploy generic systems in beam-lines or adding subsystems once the machine was already operating (insertion devices). A lot of work has been saved and a lot of extra information became available to users.

REFERENCES

[1] D. Einfeld, “Progress of ALBA”, EPAC-2008, Genoa, Italy

[2] D. Fernández et al. “Alba, a Tango based Control System in Python”, ICALEPCS'09, Kobe, Japan.

[3] A. Götz, E.Taurel, J.L. Pons, P. Verdier, J.M. Chaize, J. Meyer, F. Poncet, G. Heunen, E. Götz, A. Buteau, N. Leclercq, M. Ounsy, “TANGO a CORBA based Control System”, ICALEPCS'03, Gyeongju, Korea

[4] http://www.tango-controls.org

[5] D. Beltran, et al. “ ALBA CONTROL & CABLING DATABASE ”, ICALEPCS'09, Kobe, Japan.

[6] D. Fernández et al. “Personnel protection, equipment protection and fast interlock systems.”, Icalepcs'11. Grenoble, France

[7] S. Rubio et al., “Dynamic Attributes and other functional flexibilities of PyTango”, ICALEPCS 2009, Kobe, Japan