

# SCALING UP THE ALBA CABLING DATABASE AND PLANS TO TURN INTO AN ASSET MANAGEMENT SYSTEM

I. Costa<sup>†</sup>, A. Camps Gimenez, R. Cazorla, T. Fernández Maltas, D. Salvat\*  
ALBA-CELLS Synchrotron, Barcelona, Spain

## Abstract

The “Cabling and Controls Database” (CCDB) is a central repository where the different teams of ALBA manage the information of installed racks, equipment, cables and connectors, and their connections and technical specifications. ALBA has modernized this web application for sustainability reasons and fit new needs detected throughout the last years of operation in our facility. The application has been linked to Jira to allow tracking problems in specific installed equipment or locations. In addition, it also connects to the ALBA Inventory Pools application, the warehouse management system, where the stock of physical equipment and components are maintained to get information on the life cycle of the different devices. These new features, integrated with proprietary products like Jira and Insight, aim to become ALBA's asset management system. This paper aims to describe the main features of the recent application upgrade, currently in continuous development.

## INTRODUCTION

During the ALBA's design phase back in 2006, the Management Information Systems section (MIS), under the Computing Division, started to develop the “Cabling and Controls Database” (CCDB) [1]. Since then, this web application has been used as a central repository to keep information of all racks, equipment, connectors and cables used in ALBA. At its early stage, the application was very useful for the cabling tender and the installation phase. Subsequently, other features were implemented in order to ease the maintenance of the Tango Control System [2] and the Equipment Protection System (EPS) [3] by the Controls group, who gets the data through an API or directly from the web interface.

This in-house development has been evolving for years, and in 2019 started a process of technological upgrade. In this upgrade new features have been included, such as the integration between equipment instances and Jira, and also their integration to the ALBA Inventory Pools system. These integrations aim to become a new ALBA's Asset Management System.

## CCDB UPGRADE

In origin, the “Cabling and Controls Database” was integrated in the ALBA's Intranet and built in python Plone/Zope which acted as the web interface to ease the visualization of the data stored in a MySQL database. The

CCDB provides also an API, which has also been upgraded to a RESTful system.

Along the last years of usage, the Electronics section of ALBA, managers of the application, detected new needs and proposed new features to be developed. Without a first modernization of the technology these new features could not be developed.

## Technological Upgrade

Starting on January 2019, the original CCDB application was redefined and migrated to Django Framework [4], a Python-based free and open-source web framework that follows the model–template–views (MTV) architectural pattern. Like in the old application, the new database is also a MySQL.

This upgrade was necessary to decouple the application from the outdated Plone ALBA's Intranet, following the tendency of the other applications developed by the MIS section. In the same direction, the new application's look and feel meets the style guide of the MIS' applications, based on Bootstrap [5] (see Fig. 1).

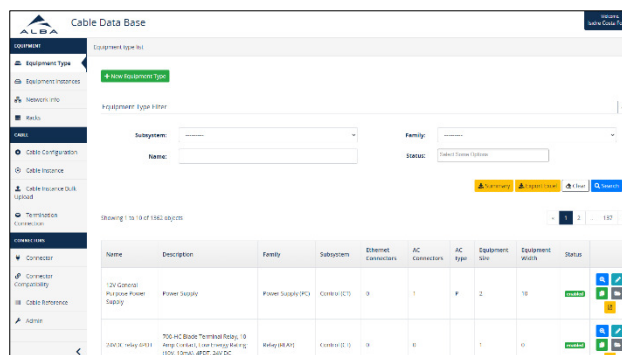


Figure 1: View of the CCDB web application.

To carry out the migration, the new application was developed from scratch and, after a period of coexistence between both the old and the new system, finally on February 2020 all the data was migrated to the new CCDB Django application and the old version was disabled.

## New Features

Among the new features introduced after the migration of the application we should highlight two of them, that enable the application to be part of a future Asset Management System.

The first of these features requested by Electronics was the tracking of equipment instances and location using Jira, our issue tracking system. This feature allows to detect dysfunctional equipment or hot locations.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2022). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

<sup>†</sup> icosta@cells.es  
\* on leave

The other requirement was the interconnection between the CCDB and the ALBA's Inventory Pools application, our warehouse management system. This integration gives users the opportunity to inform which physical equipment from the warehouse is installed in the equipment instance of the CCDB.

## ISSUE TRACKING OF EQUIPMENT

### The Concept of Equipment Instance

An equipment instance is an object created in the CCDB that represents an instance of an equipment type model. The equipment type is a template that contains a set of channels, each channel has associated a connector type that has already been created in the CCDB [1]. Furthermore, every equipment type has associated its documentation, like technical specifications and manuals.

When a user creates a new equipment instance in the CCDB, he/she is virtually placing an equipment of the chosen type in a particular location of the facility. In addition to the location and the type, user should select a system and subsystem from a list, to add a functional attribute to the equipment instance. The result of this combination of fields is an equipment code, which will be a unique identifier in the whole CCDB. In Fig.2 there is an example of equipment instance code and how it is structured.

**SR-CT-CPCI-RKA04B01-01**

**System-Subsystem-Family (Eq.Type)-Location-Code**

Figure 2: Example of equipment instance code.

### Jira Service Desk and Insight Application

In 2016 ALBA adopted Jira as its Service Management System, given that it fulfils all the needs of the organization, not only in terms of service but also in terms of project management [6]. Since the implementation of this commercial software the creation and management of different types of issues to different Service Desk teams has been centralized in a unique Customer Portal. With the support of the MIS section, every team has configured their workflows and processes according to their needs to handle their service desk requests.

As it is well known, there are a large number of plugins and applications in the market that can be installed in Jira. Some teams of ALBA started to use an application called Insight, which is fully integrated in the Jira Service Management. Insight is used for asset and configuration management [7] and it is very versatile, since any organization can completely configure its schema according to their needs.

Due to the high accessibility of the API provided by Jira, it is not difficult to make interact our own applications with Jira or Insight. However, in order to centralize all these interactions, the MIS section developed an internal REST API built in Python called jira-rest-service.

### Integration of CCDB and Jira

Regarding to the equipment instances, one of the new requirements raised by the Electronics group was to allow

users to create Jira issues from the CCDB web application relating directly the equipment instance. This feature enables the detection of dysfunctional equipment or hot locations.

The Jira issues are listed in every equipment instance detail page (see Fig. 3), or are easily reachable directly in the Jira Service Desk, where users can manage the issues at their convenience (see Fig. 4).

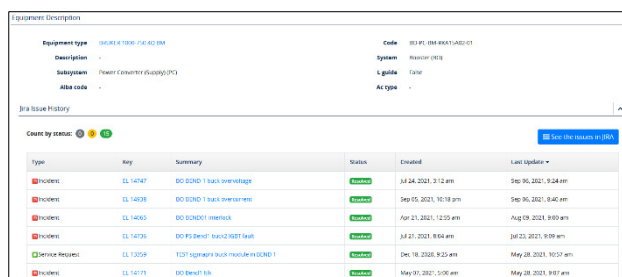


Figure 3: Equipment instance detail page with the list of its related Jira issues.

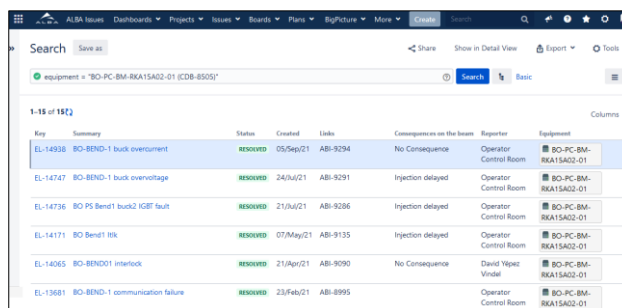


Figure 4: Jira Service Desk search page filtering by the Equipment instance showing the list of issues.

To allow this integration between CCDB and Jira, an object schema in the Insight application had to be created and configured. This schema is composed by all the possible location places that could occupy an equipment instance in CCDB, formed by combinations of fields location, number, row id and row position. Furthermore, every equipment instance created in CCDB, currently 8764 elements, are created also in Insight and related to its location place. When a new equipment instance is created in CCDB, the internal jira-rest-service API is requested to create its equivalent object to Insight, keeping the integrity of the data between both systems.

This equipment instance object of Insight is fully accessible from Jira Service Desk, since it has been defined as a custom field and added to the screens of the Electronics group Service Desk.

## INVENTORY POOLS SYSTEM

The ALBA's Inventory Pools is an in-house web application developed by the MIS section that allows the different groups of ALBA the management of the inventory of their items as well as the booking and lending of the material by other users.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2022). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

Technologically, it is built in a back-end / front-end architecture, using Django and AngularJS that interacts each other through the back-end REST services.

In a nutshell, Inventory Pools gives every group of the organization the opportunity to create their own pool, where they can manage their catalogue of materials, stocks and locations in the different warehouses or labs in the facility. To manage this information in the pool, every group assign different roles to particular users, such pool managers or pool operators, to be able to do different actions according to their responsibilities.

### Booking System

Every pool can be public or not, depending if their items can be borrowed by any member of the ALBA staff. This is one of the main features of this application, indeed this was its original motivation. Figure 5 shows a page of the booking system.

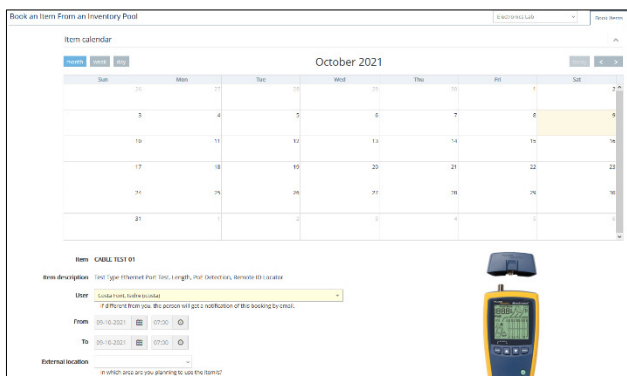


Figure 5: View of the Inventory Pools web application.

### Warehouse and Inventory Management

To ease the management of the material stock, managers and operators of every pool can set up the locations in the system where the items are actually placed in their warehouse.

Every item of the pool has its unique barcode, provided by the system or by the pool manager. This barcode can be printed in a label and posted to the item or to its location. Teams are free to let users to get the material directly from their warehouse. In this case, a set of devices (touchscreen, barcode reader and an access card reader) controlled by a raspberry pi is installed in the warehouse (see Fig. 6). User should pick up the item, read the barcode and the item will be displayed in the system. After authenticating himself with his/her ALBA's access card through the card reader he/she can get the item recording it automatically to the system. An option to search the location of the item is also available as well as an option for the pool members to manage the material directly in site.

### INSTALLATION OF ITEMS IN CCDB

As it is explained before, an equipment instance of CCDB is the representation of the specific location of the facility that will be used to place an equipment of a particular type for a determined purpose or functionality.

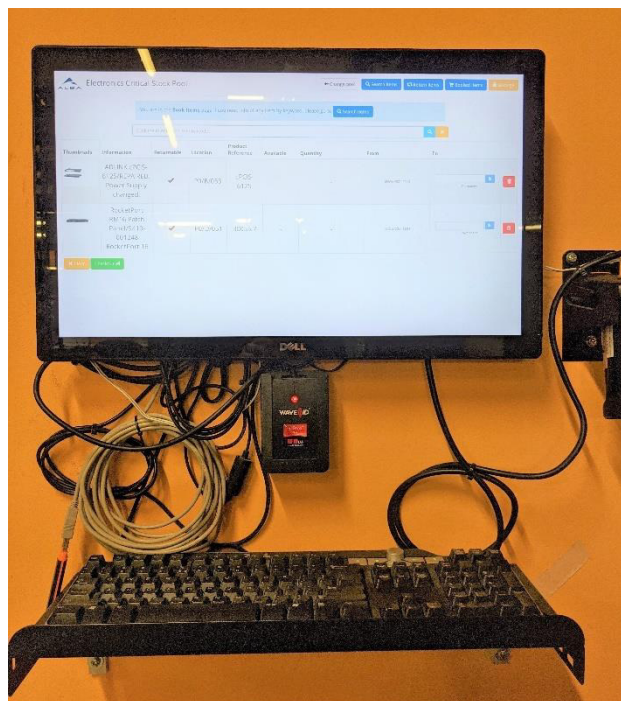


Figure 6: Set of devices installed in the ECS warehouse.

But until now, the specific physical device with its own barcode, that will occupy that equipment instance is not known. This is the need that the Electronics group detected throughout the last years of operation in our facility.

### Integration of Inventory Pools and CCDB

To fulfil this need, the MIS section has customized the generic Inventory Pools application in order to allow the interconnection of the Electronics Critical Stock (ECS) with CCDB. Thus, users can install physical devices or components from Inventory Pools to CCDB.

If a pool, like ECS, is configured to interact with CCDB some differences are applied in its features. The barcode field becomes the "Alba Code" field, with the same purpose of identifying the item. A new field "Equipment Instance" is included for each item in the list and filter page.

To install the item in the equipment instance, user is able to do it in the equipment instance of CCDB or alternatively using the warehouse feature of the Inventory Pools. For that purpose, user can go to the warehouse of ECS take the item from its location, read the barcode and inform the system that he/she is going to install it physically in the selected equipment instance of CCDB (see Fig. 7).

When user performs this operation, the item becomes unavailable for other users, because it is being used in the installation. Besides, the managers and operators of the pool can see the code of the equipment instance where the item has been installed in the list of items.

From the CCDB point of view, users can do the same operation, getting a list of items that can be installed in the equipment instance. It is allowed to install multiple items in the same instance, because an equipment instance can be composed by multiple components. In fact, there is a new feature in CCDB, still to be developed, to give the

possibility of splitting the equipment instances in sub-equipment. Doing this, users would be able to install all the components one by one in each real location within an equipment instance.

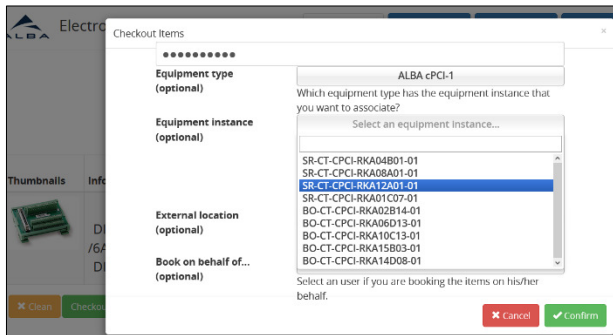


Figure 7: Example of how to select an equipment instance from the warehouse using the Inventory Pools application.

### Asset Product Object in Insight

In the integration between CCDB and Inventory Pools the same CCDB object schema of Insight has been used. A new object called Asset Product has been created. When a manager or operator of the ECS pool creates a returnable item, this item can be created to Insight as an Asset Product. Internally, this time from Inventory Pools, the jira-rest-service API is requested to create the new object in Insight. The only data necessary to do it is the Alba Code and the item’s name.

An item is considered returnable when it is a non-fungible material, when it is unique and identifiable with its own serial number. An Alba Code will be assigned to it in our system, which will be the code used to be identified also in CCDB.

When an item is installed in an equipment instance, regardless where this operation is being done, from CCDB or Inventory Pools, the jira-rest-service API will relate both objects in the Asset Product of Insight. On the opposite, an item can be uninstalled from the equipment instance in CCDB, in this case our API will remove their relationship. In Insight, an historic of events is stored for every object, so a tracking of operations in the Asset Product can be obtained.

## ASSET MANAGEMENT

The integration between both applications, CCDB and Inventory Pools with Jira and Insight can be considered as a first step to build a new Asset Management System, since the combination of these systems covers all the four aspects that every asset needs to be unique (functional, product, geographical location and life cycle) (see Fig. 8).

**Functional Aspect** This aspect gives information about what is the functionality or purpose of the asset. Normally, an asset belongs to a system and sub-system that can be translated as its functional position. This is a mandatory information for the creation of the equipment instances in CCDB.

**Product** Information about what type of product the asset is. Types can be grouped by families to ease the categorization of products. Also, the serial number or barcode, that allow the identification belong to this aspect. Stored in both CCDB and Inventory Pools there are information and documentation about the item type. While the barcode, the serial number and other manufacturer information are maintained in Inventory Pools.

**Geographical Location** Regarding to this aspect, when an equipment instance is created in CCDB, user should provide a location of the facility, formed by the combination of fields location, number, row id and row position. This combination is the part of the equipment instance code related to its location (i.e. RKA12A01).

**Life Cycle** The integration with Jira can help users to manage the life cycle of items, keeping the history of changes or problems throughout its usage in the facility.

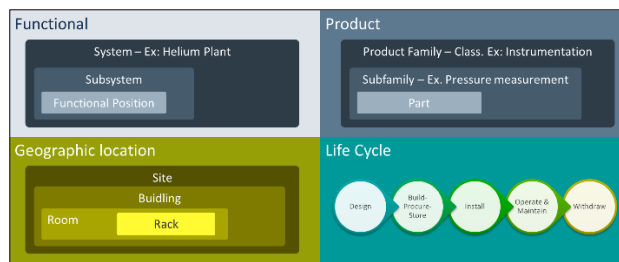


Figure 8: The four aspects of an asset [8].

### Future Roadmap

The base of this incipient asset management system has been implemented successfully, but there are new features that need to be developed in order to complete it.

First, a definition of the process of a standard life cycle of an asset is needed to be implemented in Jira and Insight. Now, users can only install or uninstall items in the equipment instance, but the life cycle is much more than this.

And also, the multi-equipment feature in CCDB is needed to provide a complete break-down of components in each equipment instance.

## CONCLUSION

The “Cabling and Controls Database” (CCDB) has been migrated to a new system for sustainability reasons, but also new features have been implemented. One of them is the integration with Jira in order to allow users the issue tracking of the equipment instances which can ease the detection of dysfunctional equipment or hot locations. Other remarkable issue is the integration with the Inventory Pools application, which enables the installation of physical equipment to the CCDB equipment instances.

These integrations are the first steps to build a new asset management system that combine the functional and geographical information provided by CCDB, the detailed information of the products from items of Inventory Pools and the tracking of the asset extracted from Jira and Insight.

There are still some features needed in order to make it more functional. Since the migration was finished, CCDB is in continuous development, not only from the asset management point of view.

## REFERENCES

- [1] D. Beltran *et al.*, “ALBA Control & Cabling Database”, in *Proc. 12th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS’09)*, Kobe, Japan, October 2009, pp. 423-425, paper WEP009.
- [2] S. Rubio-Manrique *et al.*, “A Bottom-up Approach to Automatically Configured Tango Control Systems”, in *Proc. 13th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS’11)*, Grenoble, France, October 2011, pp. 239-241, paper MOPMN003.
- [3] A. Rubio *et al.*, “ALBA Equipment Protection System, Current Status”, in *Proc. 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS’17)*, Barcelona, Spain, October 2017, pp. 1599-1603.  
doi:10.18429/JACoW-ICALEPCS2017-THPHA096
- [4] The Django Framework,  
<https://www.djangoproject.com/>
- [5] Bootstrap, <https://getbootstrap.com/>
- [6] M. Martin *et al.*, “Streamlining Support and Development Activities Across the Distinct Support Groups of the ALBA Synchrotron with the Implementation of a New Service Management System”, in *Proc. 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS’17)*, Barcelona, Spain, October 2017, pp. 298-303.  
doi:10.18429/JACoW-ICALEPCS2017-TUMPL02
- [7] Insight Application,  
<https://confluence.atlassian.com/servicemanagementserver/getting-started-with-insight-1044784303.html>
- [8] M. Martin, “Learnings from CERN on Asset Management”, Internal report, November 2018.