THE EVOLUTION OF COMPONENT DATABASE FOR APS UPGRADE[*]

D.P. Jarosz[†], S. Veseli, N.D. Arnold, J. Carwardine, G. Decker, N. Schwarz,
Argonne National Laboratory, Argonne, IL 60439, USA

Abstract

The purpose of the Advanced Photon Source Upgrade (APS-U) project is to update the facility to take advantage of the multi-bend achromat (MBA) magnet lattices, which will result in narrowly focused x-ray beams of much higher brightness. The APS-U installation has a short schedule of one-year. In order to plan and execute a task of such complexity, a collaboration between many individuals of very diverse backgrounds must exist. The Component Database (CDB) has been created to aid in documenting and managing all the parts that will go into the upgraded facility. After initial deployment and use, it became clear that the system must become more flexible, as the engineers started requesting new features such as tracking inventory assemblies, supporting relationships between components, and several usability requests. Recently, a more generic database schema has been implemented. This allows for the addition of more functionality without needing to refactor the database. The topics discussed in this paper include advantages and challenges of a more generic schema, new functionality, and plans for future work.

GENERIC DATABASE SCHEMA

The initial version of the CDB software [1] was based on a database schema designed around concepts like "component" and "design" (a design being made up of several components). As new use cases emerged for tracking and managing different sets of objects, it became clear that a new set of similar database tables had to be created every time there was a need to support a new type of object. Illustrated in Fig. 1 is a simplified conceptual design representation of how the set of tables had to be recreated for components, component instances, designs, and design elements. All of these entities had a set of standard attributes, such as a name, ownership information, description, properties, logs, and some have additional connections. This means that every time a new entity was added, it needed a set of standard associative tables to go along with it as well as any additional entity specific tables.

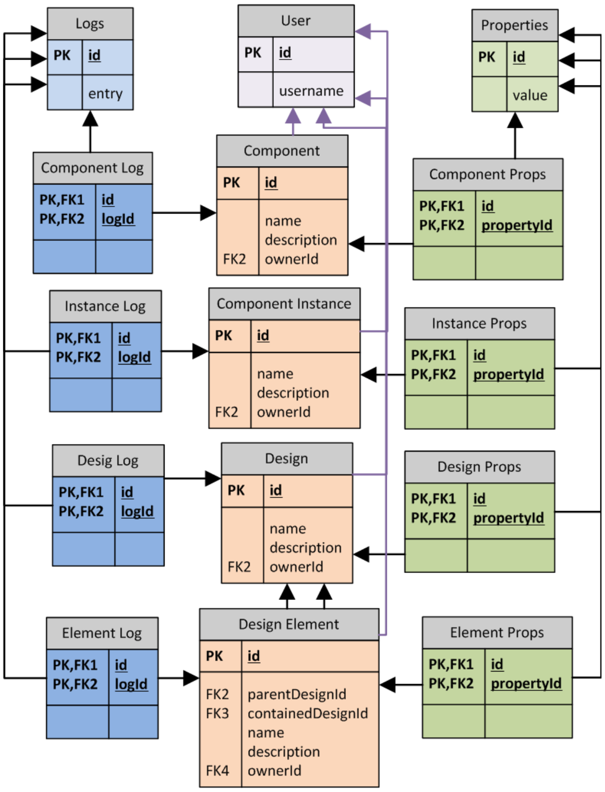


Figure 1: Conceptual simplified example of the old non-generic schema design.

Moving to a generic schema resulted in having generic tables for items and item elements (see Fig. 2). Each item now has a "self element" that represents the connections of that item. An item may also have a hierarchical structure by having more elements. All items are assigned a domain, which are similar to the previous component, component instance, etc. The assigned domain determines the possible associations of a particular item.

Advantages

The main benefit of the new schema design is that it fully supports existing application functionality in a completely generic approach. This enables us to quickly and easily add support for tracking and managing new types of objects without changing the underlying database structure.

Another aspect of the new approach is support for generic relationships. In the old schema, location could be specified only for component instances. Location information was stored in a specific location entity. A foreign key for location was referenced from the component instance table. The new schema does not have a location table, instead it has a location domain. The same required information is

[*] The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan. http://energy.gov/downloads/doe-public-access-plan

[†] djarosz@aps.anl.gov

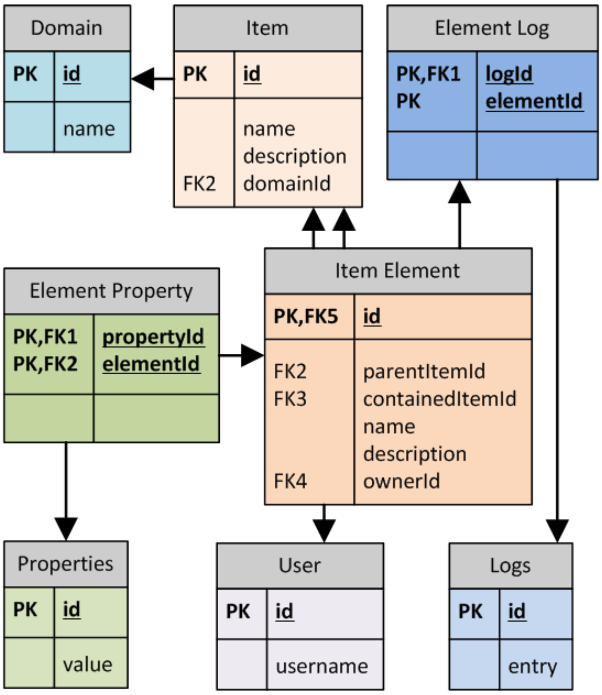


Figure 2: Conceptual simplified example of the new generic schema design.

stored for locations within items of the location domain. In order to assign an inventory item (previously known as component instance) a location, the location relationship needs to be specified between the "self elements" of the inventory item and the location item. This generic relationship provides much potential without the need to edit the database schema. Any item or item element may have some kind of a relationship. Currently only the location relationship is used, however, there are plans to add information about how items are interconnected.

NEW FUNCTIONALITY

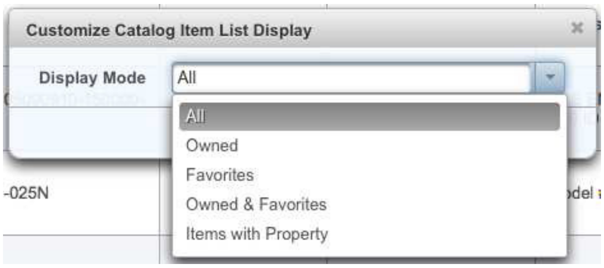
Plugin Framework

As part of the process for preparing the Component Database to become an open source project [2] we developed a plugin framework designed to provide a way of integrating CDB with external systems, and to allow developers at other facilities to add new functionality to various parts of the CDB web portal. At the APS, one use of this functionality is to provide an interface that allows users to search a separate drawing repository, and also to add an item into CDB by automatically pulling information directly from this repository. A direct link to the procurement system has also been added as a plugin.

Custom Views

The Component Database was always a highly customizable system. Users could always modify which object properties are relevant to them and which ones need to be displayed in a given view. Custom views now take this functionality to a new level. Users now have the option of specifying "favorites" or to display only items that have a certain property. These options can be saved to become the default view for that user. The described display options are shown in Fig. 3.

All customizations that are available to users are also available to groups. A user (with the appropriate permissions) can create a customized view for an entire "group". Users can select between their personal view or predefined group views.


Customize Catalog Item List Display

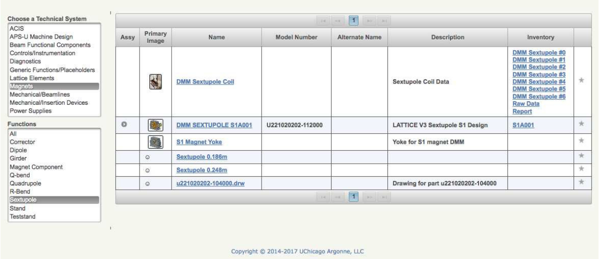
Display Mode: All

All
Owned
Favorites
Owned & Favorites
Items with Property

Figure 3: Display options for item lists.

Browse Pages

The new Browse pages allow intuitive views of the components based on their technical system, generic function, ownership, or location. This approach is helpful when trying to find components that you are only somewhat familiar with. For example, the "Browse By Function" screen, as shown in Fig. 4, allows the user to select one or more technical systems. As the user selects technical systems, the box below (Functions) populates with relevant functions and the primary list of items shows up as the user selects each of the options. The last column is inventory items. It allows users to quickly jump to an inventory item after finding a relevant catalog item.


Choose a Technical System
ACIS
APS-U Machine Design
Beam Functional Components
Controls/Instrumentation
Diagnostics
Generic Functions/Placeholders
Lattice Elements
Magnets
Mechanical/Beamlines
Mechanical/Insertion Devices
Power Supplies

Functions
All
Corrector
Dipole
Girder
Magnet Component
Q-bend
Quadrupole
R-Bend
Sextupole
Stand
Teststand

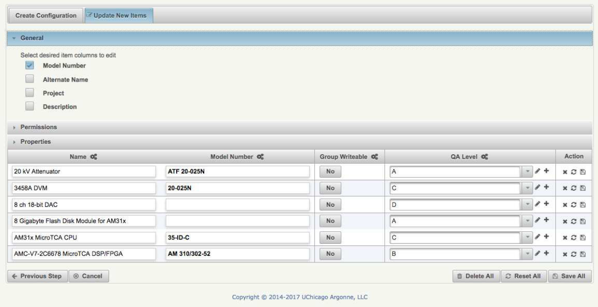
Assy	Primary Image	Name	Model Number	Alternate Name	Description	Inventory	
		DMM Sextupole Coil			Sextupole Coil Data	DMM Sextupole #0 DMM Sextupole #1 DMM Sextupole #2 DMM Sextupole #3 DMM Sextupole #4 DMM Sextupole #5 DMM Sextupole #6 Raw Data Report	★
		DMM SEXTUPOLE S1A001	U221020202-112000		LATTICE V3 Sextupole S1 Design	S1A001	★
		S1 Magnet Yoke			Yoke for S1 magnet DMM		★
		Sextupole 0.186m					★
		Sextupole 0.248m					★
		u221020202-104000.drw			Drawing for part u221020202-104000		★

Copyright © 2014-2017 UChicago Argonne, LLC

Figure 4: Browse by function page.

Multi-Edit Multi-Create Pages

Managing all of the components for a facility such as the APS is not a small task. Frequent users of the CDB requested screen views in which they could create or edit several items at once (e.g. similar to a spreadsheet). In response, the multi-edit and multi-create pages were developed which allow several items with their respective properties to be displayed/edited in a table. The user can add other columns for properties that require editing as a group. "Self-filling" of sequential parameters is supported for certain properties, where the user specifies a prefix, sequence value, and postfix. Rows of the table are then auto-filled with the prescribed sequence. The interface of the described functionality is shown in Fig. 5.



Select desired item columns to edit

- ☑ Model Number
- □ Alternate Name
- □ Project
- □ Description

Name	Model Number	Group Writeable	QA Level	Action
20 kV Attenuator	ATF 20-025N	No	A	
3458A DVM	20-025N	No	C	
8 ch 18-bit DAC		No	D	
8 Gigabyte Flash Disk Module for AM31x		No	A	
AM31x MicroTCA CPU	35-ID-C	No	C	
AMC-V7-2C6678 MicroTCA DSP/FPGA	AM 310/302-52	No	B	

Figure 5: Multi-Edit screen with a few catalog items.

Integration with FRIB Traveler System

Since CDB is the system where all the components for the upgraded facility will be entered, it is now integrated with an electronic traveler system from FRIB [3] to aid with performing various procedures on those components. The traveler system is divided into templates and traveler instances. A template is the definition of what needs to be filled out. The traveler instance on the other hand is the fillable traveler made from a template for a specific unit to be tested. In CDB, the engineer would assign traveler templates to a particular catalog item (as a property of the catalog item). When items are procured, traveler instances would be created and assigned to each specific unit (as a property of the inventory item).

The integration between the CDB and the FRIB Traveler application allows CDB users to perform many actions in the traveler system without leaving the CDB application. For example, a user can create a traveler instance from a template, assign it to an inventory item, get status information, or update details of a traveler instance (such as name, description, approval of completion of a particular traveler).

FUTURE PLANS

We will take advantage of the new generic schema and add two new domains: the Machine Design domain that will contain all components and their location for the MBA accelerator, and the MAARC domain (Measurement and Analysis Archive) for capturing characterization data and correlating it to CDB items.

In addition, we are integrating quality assurance properties and flags to ensure consistent tracking of quality control processes through the fabrication phase of components.

ACKNOWLEDGEMENTS

We would like to thank John Zientek, Jeff Toeller, Thomas Barsz, Herman Cease, Ben Stillwell, David Boughton, and other users who helped us get to where we are today with the CDB by sharing their expertise and use cases.

CONCLUSION

A significant progress has been made since the original ICALEPCS 2015 paper [1]; however, there is still a long way to go in terms of getting the CDB to be a useful tool for everyone at the APS. The generic database schema has proven to be very flexible in supporting much of the new requirements that are gathered from our users.

Since the CDB is now open source and provides a plugin framework, other organizations can easily start using and contributing to it. This could result in a valuable collaboration where the CDB would become an even more useful tool for all organizations that choose to use it.

Based on requests from CDB users at the APS, plenty of new functionality has been added, which hopefully will impact even more users. Based on different usages of CDB, there are still plenty of diverse new requirements to learn about and implement for our users.

REFERENCES

[1] S. Veseli, N.D. Arnold, J. Carwardine, G. Decker, D.P. Jarosz, N. Schwarz, “Component Database for APS Upgrade”, in *Proceedings of ICALEPCS 2015*, Melbourne, Australia, October 17-23, page THHC2O02, 2015.

[2] AdvancedPhotonSource/ComponentDB GitHub Respository, `https://github.com/AdvancedPhotonSource/ComponentDB`

[3] Frib travelerwebsite, `http://openepics.sourceforge.net/traveler/`