DESIGN AND IMPLEMENTATION OF THE CEBAF ELEMENT DATABASE *

T. Larrieu, M. Joyce, C. Slominski, JLAB, Newport News, VA 23606, U.S.A.

Abstract

With the inauguration of the CEBAF Element Database (CED) in Fall 2010, Jefferson Lab computer scientists have taken a step toward the eventual goal of a model-driven accelerator. Once fully populated, the database will be the primary repository of information used for everything from generating lattice decks to booting control computers to building controls screens.

A requirement influencing the CED design is that it provide access to not only present, but also future and past configurations of the accelerator. To accomplish this, an introspective database schema was designed that allows new elements, types, and properties to be defined on-the-fly with no changes to table structure. Used in conjunction with Oracle Workspace Manager, it allows users to query data from any time in the database history with the same tools used to query the present configuration. Users can also check-out workspaces to use as staging areas for upcoming machine configurations.

All Access to the CED is through a well-documented Application Programming Interface (API) that is translated automatically from original C++ source code into native libraries for scripting languages such as perl, php, and TCL making access to the CED easy and ubiquitous.

BACKGROUND

Previous efforts to compile a comprehensive configuration database for the Jefferson Lab CEBAF accelerator and use it to automate the setup or restoration of the electron beam have foundered on the complexity of accommodating the constantly evolving hardware installed in the accelerator and the multiplicity of elements whose properties vary with the increasing energy of each pass through the recirculating LINACS. The progress of the 12GeV upgrade project has, however focused renewed emphasis upon solving these problems by developing a central authoritative configuration database available to existing or future tools needed to operate the revamped accelerator. The resulting product has been coined the CEBAF Element Database (CED).

DATABASE DESIGN

Design

The CED has been implemented using a modified Entity-Attribute-Value with Classes and Relationships (EAV/CR) data model [1]. This design choice is appropriate because the types of information to be stored in the CED will vary from system to system. The database will necessarily evolve over time to support additional configuration attributes for existing systems as well as to model the yet-unknown parameters of the new hardware to be installed for the 12GeV upgrade.

In a traditional database schema, adding support for new accelerator hardware would involve adding additional tables and columns to the database. Such changes impose a burden by requiring software updates before the new schema can be used. In contrast, the EAV/CR data model employed by the CED is introspective – defining a new class of accelerator hardware in the CED simply involves adding rows to the already-existing metadata “catalog” tables (fig. 1). Once defined in the catalog, existing software is fully capable of interacting with the new entities after discovering their properties from the metadata tables.

Versioning

In addition to its flexibility in accommodating future data requirements, the CED schema was also designed to make use of the Oracle Workspace Manager toolkit [2]. Oracle Workspace Manager is used to version-enable the table rows in the CED, and permit simultaneous access to present, proposed, and historical versions of data.

Oracle Workspace Manager accomplishes this feat for a table by renaming it to *table name*_LT, adding a few columns to the table to store versioning metadata, creating a view on the version-enabled table using the original table name and defining INSTEAD OF triggers on the view for SQL DML operations. Clients of the CED do not need to incorporate any special logic in order to interoperate with the different versions stored in the database other than to specify which save point or workspace they wish to access.

Workspaces

A key version-enabling feature of the Oracle Workspace Manager is the ability to compartmentalize a collection of database changes into a logical construct called a workspace.

The primary workspaces is by definition the LIVE workspace and represents the current configuration of the CEBAF accelerator. Additional workspaces may be “checked out” from LIVE and used to stage upcoming configurations or to perform what-if scenarios (fig. 2). The contents of these workspaces can be merged (committed) into LIVE when appropriate.

*Notice: Authored by Jefferson Science Associates, LLC under U.S. DOE Contract No. DE-AC05-06OR23177. The U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce this manuscript for U.S. Government purposes.

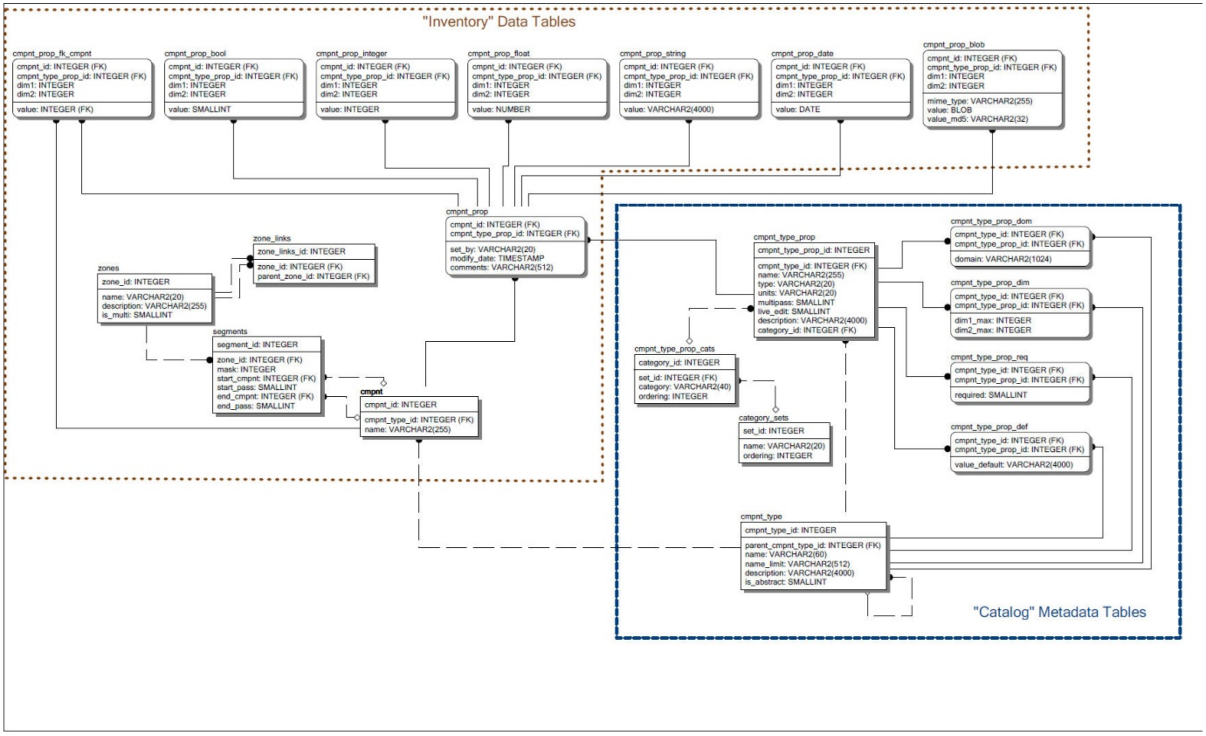


Figure 1: The core CED schema illustrating the "Catalog" tables that store the metadata defining accelerator elements. Element values stored in the "Inventory" tables as strings, floats, integers, etc. are mapped via the metadata into meaningful attributes. New types of elements may be added to the CED on-the-fly by defining their metadata – changes to the database schema are not required.

Save points

Because the Oracle Workspace Manager saves a history of all changes to a row in addition to information about its workspace membership, it is possible to query historical data simply by specifying the date and time as of which the query is being issued. To assist users, so that they don't have to specify exact dates and times, the CED provides named save points for the LIVE workspace corresponding to key configuration milestones such as the beginning or end of an experimental run or of an accelerator maintenance period (fig. 2).

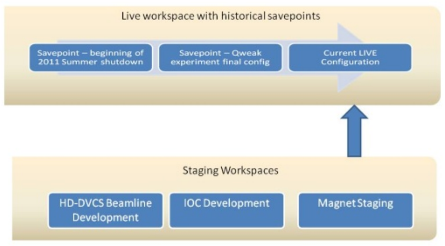

Live workspace with historical savepoints
Savepoint – beginning of 2011 Summer shutdown
Savepoint – Qweak experiment final config
Current LIVE Configuration
Staging Workspaces
HD-DVCS Beamline Development
IOC Development
Magnet Staging

Figure 2: CED Workspaces showing historical save points and staging workspaces used to prepare upcoming changes to LIVE.

In contrast to the typical approach to preserving data history using triggers and a history table containing prior values, where accessing the history requires special one-off queries or custom reports, accessing CED historical data requires no different software and no more effort than accessing the current configuration.

Configuration Control

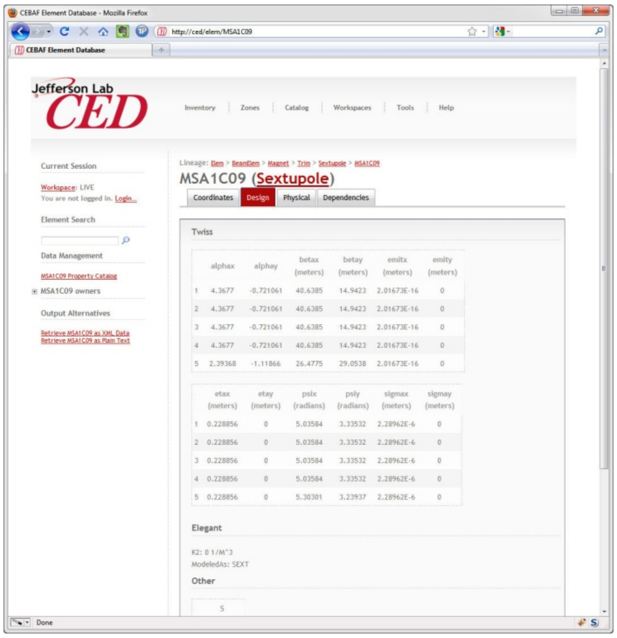
Workspaces are also used in the CED to effect a level of configuration control. With the exception of a very few properties explicitly identified as live-editable, the CED does not permit users to update data in the LIVE workspace. All updates to the CED must be performed in a staging workspace and audited for validity before being committed to LIVE.

INTERFACE

The generalized nature of the EAV/CR data model employed by CED requires a robust API to interpret the contents of the metadata tables and translate the abstract database storage into recognizable attributes useful to programs and users. The API is also responsible for enforcing much of the data validation and domain logic embodied in the metadata catalog.

For the CED, a shared library was written in C++ to be the sole interlocutor for applications that will access its information. Native versions of the API are available not only to C++ programs, but also to scripts written in Perl, PHP, and Tcl. The script language versions are generated automatically from the original library via the open source SWIG (Simplified Wrapper and Interface Generator) tool [3,4]. Having the API natively available makes it easier to update the many tools used by JLAB operators that have been written in these scripting languages over the years, as well as to develop new ones. Because they stem from the same source library, all the languages have the same functionality available and benefit from the development effort that went into testing and documenting the original compiled library.

On top of the core API, two general purpose user interfaces to the CED have also been built around the C++ library and its scripting language derivatives: a RESTful web interface and a full-featured command-line tool usable from JLAB Linux workstations.



Current Session

Workspace: LIVE
You are not logged in. Login...

Element Search

Data Management

MSA1C09 Property Catalog
MSA1C09 owners

Output Alternatives

Retrieve MSA1C09 as XML Data
Retrieve MSA1C09 as Plain Text

Lineage: Elem > BeamElem > Magnet > Trim > Sextupole > MSA1C09

MSA1C09 (Sextupole)

Twiss

	alphax	alphay	betax (meters)	betay (meters)	emitx (meters)	emity (meters)
1	4.3677	-0.721061	40.6385	14.9423	2.01673E-16	0
2	4.3677	-0.721061	40.6385	14.9423	2.01673E-16	0
3	4.3677	-0.721061	40.6385	14.9423	2.01673E-16	0
4	4.3677	-0.721061	40.6385	14.9423	2.01673E-16	0
5	2.39368	-1.11866	26.4775	29.0538	2.01673E-16	0

	etax (meters)	etay (meters)	psix (radians)	psiy (radians)	sigmax (meters)	sigmay (meters)
1	0.228856	0	5.03584	3.33532	2.28962E-6	0
2	0.228856	0	5.03584	3.33532	2.28962E-6	0
3	0.228856	0	5.03584	3.33532	2.28962E-6	0
4	0.228856	0	5.03584	3.33532	2.28962E-6	0
5	0.228856	0	5.30301	3.23937	2.28962E-6	0

Elegant

K2: 0 1/M^3
ModeledAs: SEXT

Other

S

Figure 3: CED Web Interface provides access to CED data in HTML (shown), XML, or plain text format via simple to construct URL patterns.

IMPLEMENTATION

The CED was rolled out to the JLAB community in September 2010 initially populated with data for the Magnet and Input Output Controller (IOC) systems. Data for the RF and BPM systems were added soon thereafter.

In May 2011, CEBAF entered a six months duration shutdown period during which time more than the 200 large dipole magnets were removed from the accelerator to be measured and calibrated. The CED is being used to capture and disseminate key information generated during that process, such as new field maps for the magnets, locations where shims have been installed. During the same time period, software developers and operators updated many high-level applications (such as the Interactive Elegant Explorer shown in figure 4) and control system screens to utilize the CED for their configuration when operations resume.

The focus of the CED implementation will now shift to refining the tools and processes (automated where possible) that will keep the information up-to-date and accurate. In the case of the IOC information, for example, this will include building the DHCP server configuration files directly from the CED information as well as a program that crawls and inspects the IOCs periodically to verify their configuration against the database.

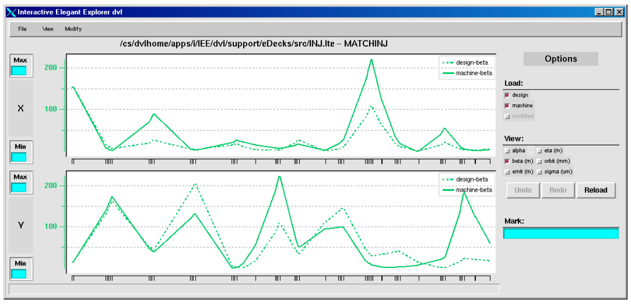


Figure 4: The Interactive Elegant Explorer now uses the CED to create a focusing lattice from a CEBAF configuration for the purpose of modeling accelerator beam transport.

REFERENCES

[1] Nadkarni, MD, Prakash M.; Marenco, MD, Luis; Chen, MD, Roland; Skoufos, PhD, Emmanouil; Shepherd, MD, DPhil, Gordon; Miller, MD, PhD, Perry (1999), "Organization of Heterogeneous Scientific Data Using the EAV/CR Representation", Journal of the American Medical Informatics Association 6 (6): 478–493, PMID 10579606

[2] "Oracle workspace Manager Overview." http://www.oracle.com/technetwork/database/twp-appdev-workspace-manager-11g-128289.pdf

[3] Beazley, David M (1996), "SWIG : An Easy to Use Tool For Integrating Scripting Languages with C and C++" Proceedings of the Fourth Annual Tcl/Tk workshop. Monterey, CA July 6-10.

[4] http://www.swig.org/