CONNECTION-ORIENTED RELATIONAL DATABASE OF THE APS CONTROL SYSTEM HARDWARE*

N. D. Arnold and D. A. Dohan
Argonne National Laboratory, Argonne, IL 60439, USA

Abstract

After the flurry of activity to construct, commission, and begin routine operation of a large user facility, the focus must then turn to long-term reliability. Key concerns include a full inventory of all installed devices, sufficient spares, quick identification of a failed device, and accurate documentation to minimize the number of system experts needed for routine maintenance.

This paper describes the Visual Connection Configuration Tool (VCCT) used in creating a searchable schema of all control system hardware. A framework is provided for identifying each installed device and its connection to the control system. The schema provides numerous benefits over a simple inventory list, such as:

- an immediate visualization of information flow through the system,
- intuitive documentation of input/output controller (IOC) hardware, subnet links and nodes, and
- a common presentation for easier cross training and maintenance.

The paper will also describe the mechanisms used to automatically populate much of the database by "discovering" the hardware through the EPICS databases and start-up scripts. Future work, such as extending the device definition to include wiring information will also be discussed.

INTRODUCTION

The Advanced Photon Source has been an operational facility since 1995. Although beam availability exceeding 97% has become the norm, there is continued emphasis on maintaining and reducing the mean time to repair and increasing the mean time between failures. Therefore, accurate and up-to-date documentation of all control devices and their configuration is imperative. Due to frequent upgrades, system enhancements, and the inevitable loss of personnel, maintaining up-to-date documentation for over 240 IOCs is difficult to achieve with conventional approaches that rely on "revision controlled drawings." This paper describes a central, searchable database of the control system devices installed at the APS, which promotes accuracy and provides online accessibility of the currently installed hardware configuration.

* Work supported by U.S. Department of Energy, Office of Basic Energy Sciences, under Contract No. W-31-109-ENG-38.

DISCOVERING INSTALLED DEVICES

In the EPICS control system architecture [1], accelerator applications, or 'databases' are developed by instantiating 'records' from a relatively small palette of record types. Each record type is made up of a set of named 'fields,' whose values are configured during the process of building an application. These fields, available to external applications, are often referred to as process variables, or PVs. The granularity in the EPICS record structure permits one to 'inspect' database code and to graphically display the application logic.

Record types that obtain data from (or write data to) external hardware use a standard hardware address protocol to specify input and output connections. Hardware addresses normally begin with the '#' character, followed by a series of character/value pairs to fully designate the appropriate field bus parameters. Each field bus type has a different pattern in the hardware addressing scheme. In the following EPICS code snippet:

```
record(ai,"L3:DG5:aDelayAI") {
   field(DESC,"Chan A Delay")
   field(DTYP,"DG535 Delay Generator (GPIB)")
   field(INP,"#L1 A15 @3")
}
```

the DTYP field indicates that a DG535 device is connected to the control system, and the hardware address pattern indicates that it is a GPIB device. By parsing this record, one infers the existence of the GPIB daisy chain (link 1) and a GPIB scanner device, which provide the data path from the DG535 to the IOC.

Another source of device information is the EPICS 'dbior' report. From the following partial report:

```
ioclic2> dbior
Driver: drvDNbug
DirectNet PLC via BUG at #L0 N12 P0 S1
DirectNet PLC via BUG at #L1 N21 P0 S1
```

one can infer the existence of two DirectNet PLC devices, connected via two BitBus links (links 0 and 1), to a BitBus VME scanner.

The entire set of EPICS records in the APS control system has been put into a PV relational database, using the Oracle RDBMS [2]. Using this exhaustive source of record/field data, an extensive data-mining process was undertaken to "discover" the APS control equipment referenced by PVs. Other sources of device information can be found in configuration statements within the start-up scripts. Using these tactics and a few heuristic rules about standard devices (e.g., every VME chassis has a power supply), it was possible to automatically "discover" approximately 75-85% of installed devices.

DEFINING A CONNECTION HIERARCHY

Physical information originating in an external sensor follows a deterministic path from the sensor to a specific IOC. This deterministic path of data flow implies a hierarchy of device connections with each device having a parent device and <potentially> several children devices. This connection orientation forms the heart of the VCCT data model and provides dramatically more information than a simple device inventory list.

Device Types

For the purpose of this paper, a "device type" is defined as a unique type of unit replaceable hardware component (e.g., card, chassis, instrument, CPU, etc.) installed at the APS as part of the control system. A device type may support information flow (e.g., a VME ADC card, a serial link, VME backplane, etc.), provide power to other components (e.g., a replaceable power supply), or simply house other equipment (e.g., an electronic enclosure).

To manage the wide range of devices at APS, each of the device types is formally described in a device type table. Unique device names are assigned to each device type, so that users interacting with the system may use only device names defined in the device type table.

Several attributes defined for each device type are quite common and need not be discussed here (e.g., manufacturer, form factor, description, function). Two particular attributes, however, are critical to the connection-oriented model applied here.

Device Interfaces

The fact that each device is defined by its connection to the control system requires device attributes related to its connection. Each device type has an attribute 'Interface Required' (IFR) that determines the types of parent devices the device can connect to. A complementary attribute, the 'Interface Provided' (IFP), restricts the types of child devices that may be connected to a device. Each device type may have more than one IFR or IFP. For a valid connection, one of the device's IFRs must match at least one of the IFPs of the potential parent device. As a simple example, it is not possible to plug an Allen-Bradley I/O card into a VME chassis because the IFR needed by the A/B cards does not match any of the IFPs exported by the VME chassis.

A helpful question that assists in working with this connection scheme is to ask for each device "What does it plug into?" (its IFR) or "What can plug into it" (its IFP).

Device Table of Device Instances

Each device installed at the APS is an instance of one of the defined device types. To date, 13,384 devices have been identified and entered in VCCT. Additional attributes required to fully describe an installed device (a device instance) include its location, serial number, logical number (used by software to identify a particular device), and physical slot (a physical slot number or position in a daisy chain). To track the connection-oriented hierarchy, a parent-id attribute is required that is a reference to another device that is this device's parent. Each device supplies data to (or receives commands from) or supplies power to its parent device.

Many of the device attributes are optional and are placeholders until valid data may be entered.

The Physical Data Model for VCCT shown in Figure 1 illustrates the relationship between the device_table, and the device_type_table and its related tables.

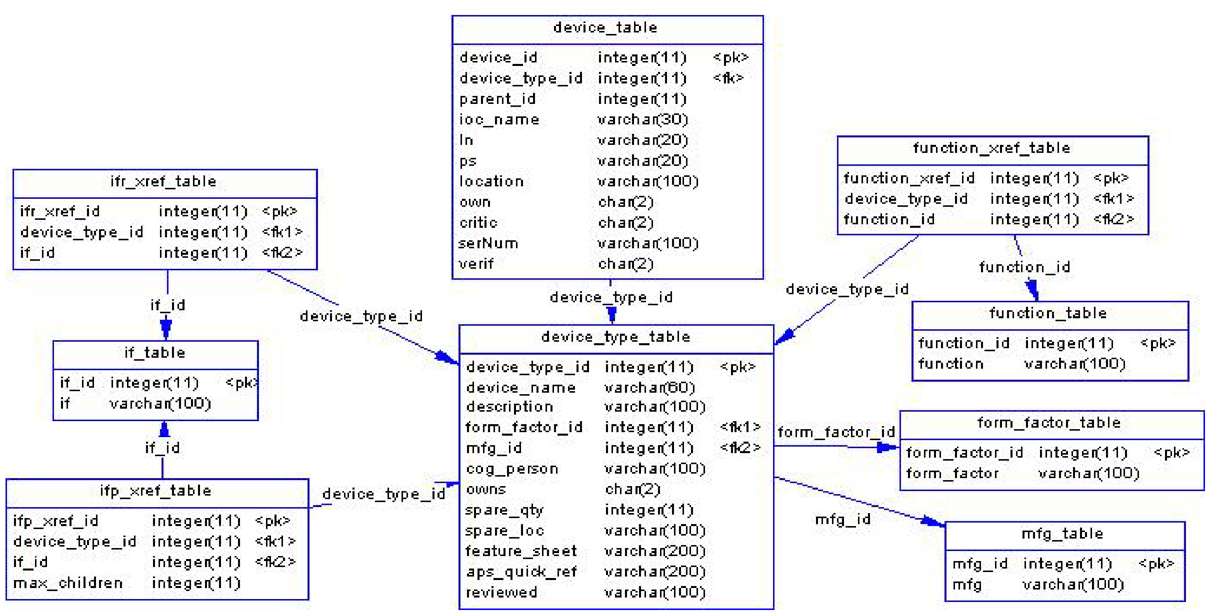

Figure 1: VCCT Physical Data Model.

DEVTREE

The data mining process was successful in 'discovering' the vast majority of the APS control system equipment. To provide the ability to manually complete, update, and display the device hierarchy, a graphical user interface (GUI) tool 'DevTree' was developed. DevTree was developed in Java to allow for platform independence. For the initial implementation, the physical data model shown in Figure 1 was implemented using a flat file for each IOC, to minimize collisions between multiple user editing sessions (see the following section for plans on using MySQL).

DevTree allows the user to interact with the connection hierarchy much like a file system hierarchy. Figure 2 shows an example of the DevTree display GUI.

If a device is selected in the connection hierarchy, then details of its location, logical number, physical slot, etc. are shown in the top panel. The user readily sees all the devices that share any backplane or field bus daisy chain and can immediately observe the data flow from the IOC to the instrument. Branches may be expanded or collapsed to adjust the view as desired.

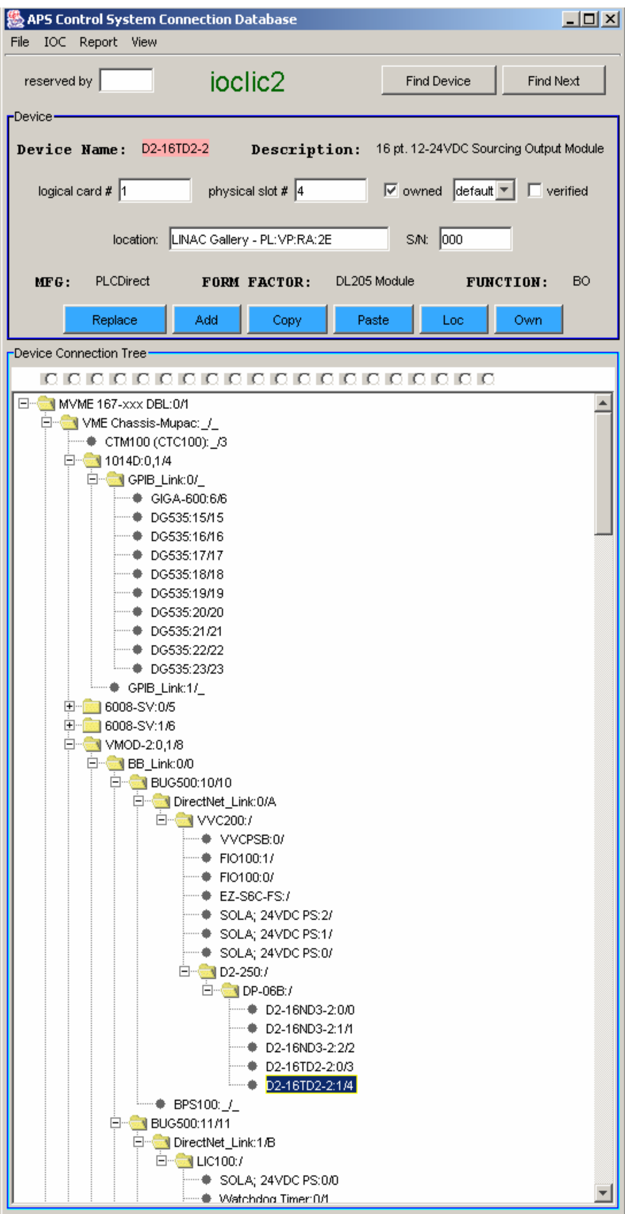

Figure 2: VCCT Graphical User Interface.

When the user wants to add a device to the hierarchy, only those device types whose IFR matches the selected device's IFP will be displayed. This minimizes configuration errors by not allowing inconsistent connections (e.g., you cannot plug a VXI board into a VME chassis). A string entry field is provided to filter the list of choices should it be too long.

FUTURE PLANS

- DevTree is now used exclusively to maintain the searchable control equipment inventory. An effort is in progress to convert VCCT to using a relational database. The MySQL database has been set up using the physical data model of Figure 1, and the VCCT flat file data has been imported into the relational database. This information will greatly enhance the accessibility of the data for more advanced reporting and searching.
- Working with the connection hierarchies has revealed that devices actually belong to three hierarchies at once: control flow, power, and housing. Work is in process to better define this tri-hierarchy schema, which will lead to a more thorough device definition.
- Discussion has begun to extend this idea to allow full documentation of interdevice wiring.
- We recognize that the connection-oriented hierarchy has the potential to allow tracing of a fault from a PV to the device where the information flow is broken. Another GUI tool that interacts with this database, the PV database, and live PV data (via Channel Access) to provide this feature has been envisaged.

CONCLUSION

Maintaining accurate information on more than 13,000 installed devices in the APS control system presents a huge challenge. Implementing a connection-oriented approach to defining installed equipment has made significant progress in addressing this problem. The convenience of a GUI tool (DevTree) to maintain this hierarchy promotes wide cooperation and accurate entries. This groundwork promises a huge payback as more tools are developed to take advantage of this data.

REFERENCES

[1] L.R. Dalesio, M.R. Kraimer, A.J. Kozubal, "EPICS Architecture," ICALEPCS 1991 Proceedings, C.O. Pac, S Kurokowa, and T. Katoh, Eds., (ICALEPCS, KEK, Tsukuba, Japan, 1991), pp. 278-282.

[2] D. A. Dohan, "Decision Support System for the APS Control System," ICALEPCS 2001, San Jose, November 2001, pp. 271-273 (2002).