INTEGRATED RELATIONAL MODELING OF SOFTWARE, HARDWARE, AND CABLE DATABASES AT THE APS*

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

Abstract

The APS control system network of ~240 IOCs is implemented using the EPICS database/record/field software architecture. Low-level IOC logic has been modeled and captured in a global relational database [1], providing the users with a variety of inter-IOC query capabilities. This coverage has been extended to include client applications connected to EPICS process variables. A recent effort broadened the relational database coverage to include the control system hardware [2], in which the devices associated with each of the IOCs have been captured in a 'connection' hierarchy. Along with the equipment detail, information related to the flow of information from the device to the IOC is stored. This provides an immediate visualization of information flow through the system and its field buses, and facilitates a convenient 'drill down' mechanism to locate any device.

The present work describes the enhancement of this device connection database to include other necessary hierarchies. In addition to the original 'control flow' hierarchy, any given device may now be characterized by how it is physically housed (housing hierarchy) and how it derives its power (power hierarchy). This allows equipment not directly accessible by EPICS software (e.g., a power supply) to be included in the database.

The final control system domain to be modeled consists of the many complex cabling systems that integrate and synchronize the operation of the IOC devices and their accompanying software databases. These cabling systems include the device I/O cabling, the timing system, and cable subsystems such as the machine status link and the machine protection system. Difficulties in managing the inherent complexity[1] of these systems and issues concerning graphical presentation of the information will be discussed.

INTRODUCTION

A typical accelerator control system is a complex integration of chassis, modules, instruments, hardware components, cable plants, field wiring, process-variable databases, and ancillary application programs all working together in unison. A change to any element typically requires a coordinated change in other elements (e.g., adding an input module also requires adding field wiring and databases). Likewise, failure of an element will cause other related elements to operate improperly (e.g., a power supply failure can affect numerous devices).

Typical documentation schemes utilizing "revision controlled documents" are not practical for managing such a complex system. Defining element dependencies is much more suited to a relational database implementation of documentation rather than a drawing-based method. Our current project is an attempt to fully document an accelerator control system using relational database technology. If successful, the benefits will allow us to:

- immediately trace faults back to the root cause,
- instantaneously update installed inventory statistics,
- locate all installed devices of a particular type,
- correlate a failed device with the application software that uses it,
- identify exactly what will fail if a specific cable is removed (or a particular breaker trips, or a particular module fails),
- provide graphical user interfaces to all the above.

CONTROL EQUIPMENT DATABASE

Rather than modeling control system components as a complex hierarchy of classes and subclasses, the APS schema defines a device simply as a unit-replaceable component that accepts control commands or provides data to (i.e., communicates with) the control system. The instance detail for each device is stored in a single device table that has a reference to a device-type table, which stores the information common to all devices of a given type. Each entry in the device table has a reference to a 'control-parent' device that is the device to which it delivers control data. Data passing up the hierarchy eventually reaches the root of the hierarchy, the IOC CPU. All IOCs have an implied common root–the site Ethernet.

The database thus contains not only device instance and type information, it also contains information concerning the topology of control system data flow. In essence, the database models 'control connections,' with the device detail (its type, slot, manufacturer, etc.) treated as attributes of the connection. This connection-oriented schema provides an intuitive method with which the user interacts with the database when locating a particular device or adding new components.

As a measure of the complexity of the APS control system, there are more than 600 distinct device types and over 14,000 installed device instances. Adding new device types is simply a matter of inserting entries to the device-type table and does not require modifying the underlying schema.

Recognizing that the value of the data in the database is only as good as its completeness and accuracy, an independent verification effort of the database contents versus the installed devices is imperative. However, verifying installed hardware to a flat list of 14,000 entries is an insurmountable task and highly prone to error. The connection-oriented hierarchy of this approach, however, provides the mechanism to divide this task into the more

---

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

manageable subtasks of verifying each individual path of the hierarchy. In addition, a visual representation of the hierarchy is much more intuitive and ensures accuracy and completeness.

MULTIPLE HIERARCHIES

The schema described in the previous section addresses the modeling of those components in the control system that are directly addressable by an IOC CPU, the reference root. However, in any control system there are many infrastructure components (patch panels, power adapters, media converters, etc.) whose components are not directly addressable from an IOC and are not directly 'owned' by any specific IOC. They form a vital part of the control system and must be included in the equipment inventory.

To address these categories of devices, two additional hierarchies have been introduced into the database schema–the 'housing' and 'power' hierarchies. Each device in the housing hierarchy contains a reference to its 'housing parent'–i.e., the parent device within which the child is physically located or housed. For example, a VME chassis houses its VME modules, independent of whether there is a CPU in the chassis, as in the case of the timing or event systems. Similarly, each device in the power hierarchy has a reference to the parent device from which it derives its power.

A given device must belong to the housing hierarchy, since it must be housed somewhere, and may optionally belong to one or both of the control and power hierarchies. An example of a device belonging to all three of the hierarchies is a VME chassis, which houses, controls, and powers its member VME modules. The additional two hierarchies are integrated into the database by simply adding two attributes to the existing device database, the references to the device's housing and power parents, if appropriate.

The housing hierarchy provides a powerful method to document the location of any device. A device location is simply its housing hierarchy path, obtained by traversing the housing hierarchy from the device to the housing root. As in the case of the controls hierarchy, the housing hierarchy consists of several 'reference' roots, corresponding to the rooms that are part of the APS building complex. A typical housing hierarchy path may contain the elements: room/rack/chassis/module.

Using the housing hierarchy to extract device location information provides a number of advantages. First, the original free-form nature of the location attribute is replaced with the formal name space provided by the device database. This makes the location of a device a queryable quantity–one can request all the devices that are located in a given rack, for example. As in the case of the controls hierarchy, a second benefit is the process by which the user can verify database completeness. At the top of the housing hierarchy the user can verify that the number of racks or enclosures in any given room corresponds to the contents of the database. For each rack in turn, the user can verify the number of chassis or other rack-mounted devices in the rack. This process can be applied successively through the hierarchy.

An example view of a device residing in each of the hierarchies is shown in Figure 1.

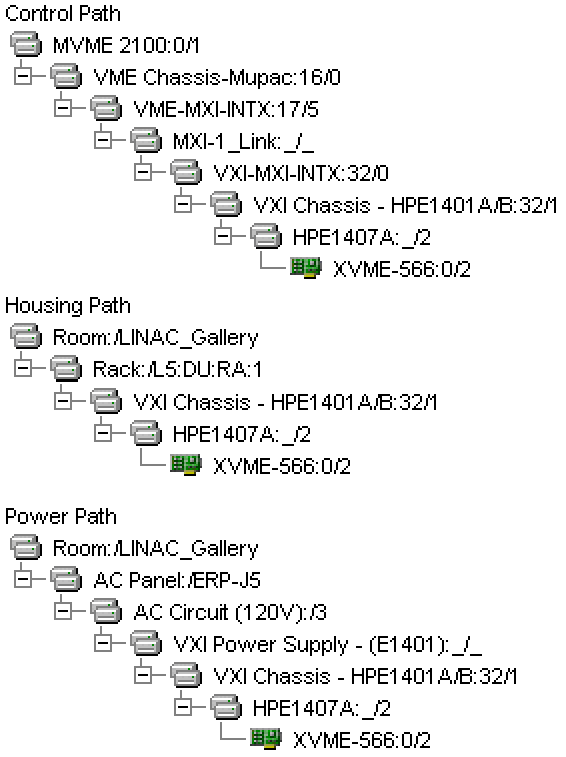

Figure 1: Multi-hierarchical device path display.

DEVICE PORTS AND THE CABLE DATABASE

The 'leaf' devices, those at the end of the device hierarchies described above, provide the interface between the control system and the external world. These devices contain 'ports' through which real-world information–voltages, currents, etc.–enters the control system.

One of the attributes of the device-type table contains the number of input/output ports for each device type. A new database table (the port table) has been added to the schema to include the device ports. Each port entry has a reference to the device containing the port, an attribute indicating whether the port is an input or an output, and the port label. When an I/O device is added to the equipment database, the database application adds the relevant number of ports to the port table.

With this infrastructure in place, it is straightforward to define the cable database. This consists of a single table (the cable table), in which each cable record simply describes the connection between two ports in the port table. The port table entry contains a link to the device that contains the port. One can use the housing and control topology information contained in the device table to completely document the cable database. The physical data model for the database schema is shown in Figure 2.

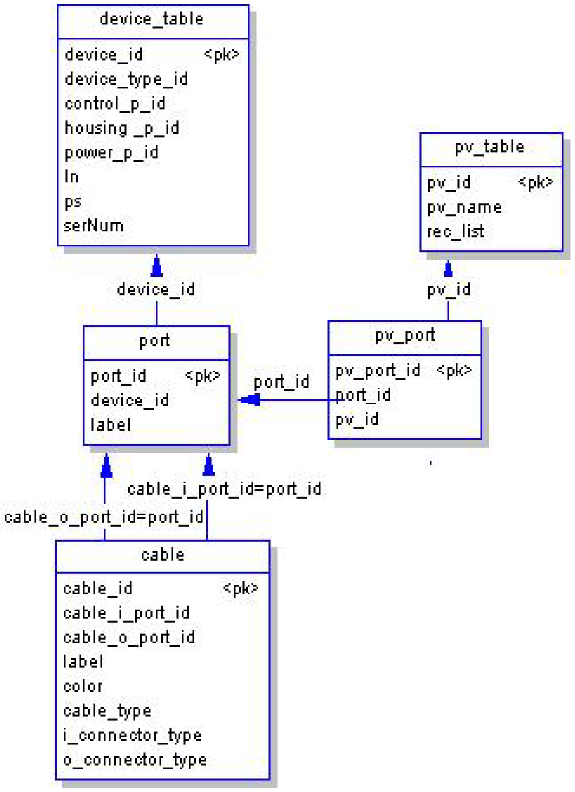

Figure 2: Physical data model for the device, port, and cabling databases.

The cable table contains additional instance data (wire color, cable ID, etc.). This instance data provides the basis for the wiring list documentation.

FUTURE PLANS

The control system cables carry ‘signals’, i.e., named information flows between device ports. These signals are closely related to EPICS process variables. This suggests the possibility of interrelating the hardware, cabling, and software databases, as suggested in Figure 2.

Preliminary work towards this goal has taken place in developing the Process Variable database [1]. That paper discusses the datamining of the process variable database to infer controls hardware devices, thus relating process variables with hardware. Work is underway to further investigate this possibility.

CONCLUSION

The device inventory based on the controls hierarchy has been in use for approximately a year. It has received wide acceptance both from the application developers, as well as from the field technicians. The intuitive hierarchical drill down mechanism for traversing through the database has contributed to the acceptance of the relational database.

Extending the database schema to handle multiple hierarchies provides the ability to capture the remaining components in the control system. It significantly extends the amount of information contained in the database, including the housing information, which provides location information that can be queried. In addition, the power distribution topology is documented, which provides valuable information for locating “problems related to the delivery of control power.”

REFERENCES

[1] D. A. Dohan, “Decision Support Facility for the APS Control System,” Proceedings of ICALEPCS 2001, San Jose, CA, SLAC-2-592, pp. 271-273 (2002).

[2] N. D. Arnold and D. A. Dohan, “Connection-Oriented Relational Database of the APS Control System Hardware,” Proceedings of PAC2003, Portland, OR, pp. 2327-2329 (2003).