# Functional Requirements for the Hardware Database System in DUNE and Options for its Design

M. Potekhin, B. Viren

October 16, 2019

**Abstract**

The DUNE experiment requires systems and tools for management, control and accounting in the areas of manufacturing, QC, logistics, installation and maintenance. It is important to ensure proper integration, interconnectivity and interoperability of these databases as well as interfaces to other systems such as calibration, slow controls etc. It is also important to ensure the possibility to describe the evolution of various hardware modules in time in terms of their composition (e.g. circuits replaced) and potentially even their structure; this implies version control capability. We propose a hierarchical approach to the description of the apparatus and a state transition model for the construction process. Accordingly, we consider a few *functional* requirements to the databases and their interfaces in the context presented above, as well as implications and options for the system design.

## 1 Overview

The most recent experience in DUNE with systems of this kind comes from the protoDUNE-SP experiment. The experiment employed several hardware/QC databases which were distinct for a few elements of the apparatus. For example, the Cold Electronics QC systems were designed specifically for purposes of testing and quality control of the novel Cold Electronics and as such provided ample capabilities for storing and analysis of the data collected. A dedicated database was used for managing the CE information.

Other databases were created (sometimes only as prototypes) to support manufacturing, transport and installation of protoDUNE-SP elements such as the PD paddles, APA/CPA and other TPC components. These were distinct from the CE DB systems and based on the platform commonly referred to as "FNAL hardware DB" [1] with separate databases used for different element types. In each such database the relationships between individual components and modules were described by reference utilizing a consistent set of primary IDs, specific for the particular DB. No common numbering scheme was used. Some information (e.g. logistics) remained external to the DB and had to be managed separately.

The DUNE experiment represents a vastly larger scale in terms of the component count, overall complexity and significantly more challenging installation procedures. Optimal execution of manufacturing, installation, commissioning and operations will require a well thought out design of the databases in the context presented here. Possible reuse and adaptation of existing technologies is highly desirable.

# 2 The Model of the Apparatus

## 2.1 Disclaimer

The model presented below is not a new concept but a generalization of design principles observed in databases such as in the FNAL Hardware DB [1].

## 2.2 Components and Aggregations

It is natural to describe the DUNE apparatus in terms of hierarchy of elements e.g. model it as a *graph*. Same applied to its components, for example a FEMB can be modeled as an *aggregation* which contains various electronics components e.g. ASICS, and can be decsribed as a graph. This approach was in fact used in the individual DB schemas for protoDUNE-SP created in the FNAL Hardware DB [1], by using foreign keys to connect related tables.

Most DUNE components are aggregations themselves, although in graph terms, the leaves of the tree are atomic by definition (i.e. an individual ASIC). Definining the levels and granularity of the hierarchical structure of components in DUNE is beyond the scope of this paper, we just note here that this will need to be done in the design phase. In the following we assume that a *component* may be either atomic or an *aggregation* of other components subject to the same assumption.

## 2.3 States and Transitions

To describe the progression of events from the design phase of the detector to its construction, installation and operation we propose the following model:

- Each component can exist in a variety of *states*, and its set of possible states will differ depending on the component class. For example, the component can be in states like [ *"ordered", "in transit", "received", "tested", "put in long term storage",...*] etc
- The chain of events starting from design/ordering/manufacture (or procurement) and then on to QC, shipping, installation, testing "in situ" etc is reflected by the changing state of the component as recorded in the database
- The exact sequence of transitions depends on the exact nature (class) of the specific component; one critically important case of transition is *aggregation*
- Each state transition may optionally result in a data product specific to the component and the specific transition. One example is metrics collected during the QC procedures for CE. The exact nature, size and format of data thus produced depends on the specific component and transition
- In the item above, there will be cases when the database will only contain *references* to external data storage and sources as it is often impractical to store certain data products e.g. ROOT files directly in the DB
- Each state transition may optionally require a record in the audit trail e.g. the time stamp, the reason for the event any comments required in the process and the entity responsible for changing the state and updating the record (a person and/or institution). Examples already exist in the "FNAL Hardware DB"

## 2.4 Evolution of Aggregate Components

Consider the use case where a particular component within an aggregation needs to be replaced, such as a failed ASIC installed on a board. This process can be modeled in at least two ways:

- The old aggregation (e.g. the board) is marked as “retired” and the audit trail record created; a new aggregation is created, assigned a new *unique ID* and assumes the place of the old one
- Or, the aggregation with its existing *unique ID* is preserved but is assigned a new *version number*; again, an audit trail record is created

The concrete choice of solution will depend on the particular component so it needs to be deferred to the design and implementation stage.

The hierarchical model presented in this paper allows to handle an important use case whereby a component (module) is completely redesigned in terms of its construction and composition while retaining its functionality and place in the overall hierarchy. Same applies to use cases where new aggregations of already existing components need to be defined for optimal management of the construction process.

## 2.5 Component Location and Ownership

We will use the concept of the *functional position* introduced in [2], which refers to a physical position of a component specific to its function and which cannot be moved. It is complementary to the component geographic location. Ownership refers to the entity who is assigned custody of the item and is authorized to perform specific actions relate to the item. We note that the geographical location, functional position and ownership of an item are an integral part of its state.

# 3 Summary of Functional Requirements

## 3.1 The Unique ID Requirement

To make it possible to create and evolve the hierarchical description of DUNE along the lines described above (2.2) we propose the following requirement:

- Every component used in DUNE — either atomic or an aggregation — needs to be *uniquely identifiable* in a way that allows creation of any type of aggregate of components
- To make this possible, the unique identifiers in this context should be distinct from the manufacturers' and other ad hoc IDs. At the same time the unque IDs *shall be associated* with the manufacturers' and other ad hoc IDs in the database

Note that the above requirement is crucial to allow for the eventual evolution of DUNE as it undergoes construction and installation. New aggregations can thus be created where necessary.

## 3.2 The Traceability Requirement

It must be possible, given the ID (and perhaps other attributes or references of a component described in the database) to identify it in documents such as technical drawings, schematics and any other documentation for QC and installation. For example, given the component ID it must be possible to identify and retrieve a document describing the relevant QC procedures. It is implied that the exact provenance (e.g. the manufacturer) of each component must be recorded, and queries based on such attributes must be possible.

## 3.3 The Completeness Requirement

To ensure the functionality of the database we propose the following requirement:

- each component shall be fully described in the system with level of detail necessary to ensure successful manufacture, QC, delivery, assembly, installation, commissioning and operation and maintenance of DUNE, and also document possible failures in any of these steps

- such information will be specific to each *component class* in any of its possible states

## 3.4 Record of State Transitions Requirements

- History: a complete history of state transitions of a component preserved in the database (where applicable)

- Timelime: a "period of validity" will be recorded for each state, and essentially defined by the timestamps of transitioning into a particular state and leaving it for the next state

- If the state transition is due to application of a manufacturing step, a specific test or some other procedure the nature and the parameters of such procedure must be recorded in the system and be *uniquely identifiable*.

- The complete set of state transitions (2.3) of a component forms its lifecycle. These may evolve changing the composition, physical characteristics, location, and associated documentation. The system must provide adequate means to manage the component lifecycle, including its testing and QC, *evolving documentation and additional metrics collected after its deployment*

## 3.5 Connectivity Requirements

To provide information on expected and actual assembly of components the following is required:

- The *expected* or *functional position* connectivity shall be modeled for both physical and electrical relationships

- The *actual* or *as-built* or *as-found* physical and electrical connectivity shall be modeled in terms that are relative to the expectation

- The *connectivity* as described in the system will be modeled according to the *state and state transition* concept as set forth in 2.3 so it will be subject to the "history" requirement as defined in 3.4

## 3.6 Interface Requirements

### 3.6.1 Network and Web

The database will be required to have a variety of interfaces as dictated by its applications e.g.

- Web interface for operators and managers, which will effectively create a full-fledged information system
- Other network-based interfaces, including command-line interface which will allow automation (such as scripting) of operations and integration of various production processes

### 3.6.2 Interfaces for the Component Lifecycle Management

As an example, there are cases like the CE QC and other test data where the metrics thus collected form a part of the initial calibrations parameters. For this reason, appropriate interfaces must be created to ensure reliable transfer of such data across the systems. In order to evaluate the performance history and condition of a particular electronic component it must be possible to get access to subsequent iterations of the calibration data as well, i.e. to provide access to data sources outside of the hardware database itself.

### 3.6.3 Authentication/Authorization

To properly manage the component lifecycle and a variety of other information presented in the database it will be necessary to integrate with a role-based authentication/authorization system e.g a site SSO.

### 3.6.4 Industrial ID Systems and Mobile Platforms

The DUNE Hardware Database will have the functionality for integration with barcode and QR scanners to facilitate all stages of the construction process wherever such devices are available. These do not serve as a substitute to the unique component ID described above but are complementary to it.

When testing large-scale components of the DUNE experiment and also in the installation stage it may become necessary to use mobile devices (tablets, scanners etc) in the field to feed information to the database and also to read data from the database. Such interfaces must be included in the design.

# 4 Reusability

It is foreseen that any design or adaptation of a system that meets the functional requirements listed above will require a substantial investment of effort and resources. For this reason, we propose to state a collateral requirement that the system used in DUNE should be experiment-agnostic in terms of its architecture, and be adaptable to other projects. This will help to assure a more efficient use of resources and may serve to attract collaboration from other experiments as a part of a cross-cutting effort.

# 5 The Timeline

Component manufacturing and construction of the DUNE experiment has at least two distinct time scales that need to be addressed in the context presented here:

- Near and Medium Term
  - Preparations for the protoDUNE-2 experiment, with manufacturing of components to commence in 2020, and installation scheduled for 2021
  - Early manufacturing of certain components of the DUNE experiment
- Longer term, including manufacturing and installation of the DUNE detector, commissioning and operation

# 6 The Proposal

## 6.1 Bridging the Gap

While there is a substantial time pressure to address the needs of DUNE in the near and medium term, at the time of writing it seems unlikely that a system capable of addressing the requirements presented in this paper can be created (or adopted from another experiment) on such short timescale. We propose to mitigate this in the following manner:

- Survey and catalog the interim systems (e.g. existing and in use at participating sites) which are likely to be used in near and medium term
- Identify additional attributes (if any) to be added to object models in these systems and modest changes in functionality to allow for eventual migration to the final system
- Define formats for data export and ingestion necessary for such migration as well as future I/O and data exchange protocols

## 6.2 Action Items

In addition to interim measures described above we propose the following action items:

- conduct a comprehensive survey of the DUNE Consortia and computing groups in order to agree on a common set of requirements for the database in the context presented in this paper
- evaluate a few different design options, such as summarized in the next section (7). The ultimate aim of this effort is to preferably identify a possible existing solution or perhaps a new deisgn satisfying the requirements of the Collaboration at large.

The key motivations are:

- to avoid duplication of effort and meet the requirements to the greatest extent possible
- to prevent divergent and/or incompatible designs in each sybsystem area and avoid operational risks associated with that
- to provide a system-wide view of the production, testing, QC, logistics and installation process necessary for efficiency of management, identification of problem areas and bottlenecks and minimizing errors in all stages of the DUNE construction

# 7 Options

## 7.1 Reuse

Possible reuse of components and/or existing systems should be considered whenever possible, with adaptations as necessary to conform to the functional requirements outlined above. We consider two candidates below, without necessarily limiting ourselves to these two.

### 7.1.1 The FNAL Hardware DB

The current version of the FNAL Hardware DB [1] is a collection of tools aimed at facilitating the creation, management and use of the hardware databases required by the groups working on manufacturing and QC of various detector components. The system was used in building two subsystem of the protoDUNE-SP experiment.

- **Pros**
  - A well understood system proven in previous experiments including protoDUNE-SP, with good support and expertise at FNAL
  - Supports many of the requirements listed above
- **Cons**
  - A somewhat rigid schema not well suited for evolution (e.g. QC procedures and logistics essentially hardcoded in the schemas created for individual components, some schemas de-normalized)
  - Still lacking a few features necessary to meet the DUNE requirements, no solid support for the component lifecycle
  - Current version of the Web UI is not well suited to support the functionality to meet the requirements discussed here
- **Reuse strategy**
  - Using experience with the system, upgrade it by introducing a Web framework allowing for OO-based modeling and schema evolution (cf. ORM)
  - Implement integration with external systems
  - Update the Web UI and other clients

### 7.1.2 ATLAS Equipment Database

An introduction to the ATLAS Equipment DB is presented in [2].

- **Pros**
  - A full featured system covering many complex use cases
  - Likely supports most and perhaps all of the DUNE requirements
  - Proven during construction, installation and operation of a very complex detector
  - Integration of a number of separate databases (using the Glance system, [3])

- **Cons**
  - Substantially based on a proprietory commercial system (Infor EAM [4], formerly d7i), entailing the licensing and lifecycle issues
  - Has many hooks into the vast CERN information infrastructure which would be hard to disentangle in the reuse scenario
  - Considerable amount of work was required to tool the database for specific ATLAS needs, this part won't be immediately reusable in any case

## 7.2 New Development

At present, creation of a new hardware database for DUNE is also on the table as a design option. Discussion of the possible design directions and cost/effort estimations relative to adoption of an existing solution are deferred till a later time.

# References


[1] *Hardware DB Wiki* `https://cdcvs.fnal.gov/redmine/projects/hardwaredb/wiki`

[2] *Management of Equipment Databases at CERN for the ATLAS Experiment*
DOI: 10.1142/9789812819093_0129

[3] *Glance project: a database retrieval mechanism for the ATLAS detector*
DOI: 10.1088/1742-6596/119/4/042020

[4] *Infor EAM* `https://www.infor.com/products/eam`