Fermilab MTF

Doc Number: MTF-92-0010
Version: 1.4
Category: DATABASE DOC

Design of the **subjects** Virtual Database

Bruce C. Brown
Fermi National Accelerator Laboratory *
P.O. Box 500
Batavia, Illinois 60510

7/13/94

*Operated by the Universities Research Association under contract with the U. S. Department of Energy

Contents

Abstract

The subjects virtual database contains information on measurement subjects, the series they belong to, and their components.

1 Introduction

In designing our measurement system we wish to minimize the number of times in which measurement personnel must enter detailed information on a given object. To facilitate this, we designate each object as a test subject. To each test subject we assign a unique serial number which will be used when recording further measurement efforts. Note that each test subject is considered a member of a series of test subject.[1]

Many of the devices which we examine are fabricated from parts which may separately be of interest to us. A connection will be maintained between the subject and the components from which it is constructed at both the individual subject and the series levels. We accomplish this at both levels by constructing a hierarchical description of the object with a pair of related tables.[2]

This design report should be examined in connection with the contents of the TOT tables[3] and the idraw picture of the subjects virtual database (available under the code management system as subject_1.idr). Unlike several other MTF virtual databases, this database is likely to be used by a wide variety of users, including engineers and physicists many who are not so familiar with the tools. For that reason, more fields have been which do not use the pre-defined field lengths (aliasstr, namestr, titlestr, descstr) but instead several use varchar(8) and one uses varchar(16). This permits a query to have a more suitable format using the default lengths.

2 Series of Measurement Subjects

The series and series components tables capture a portion of the design specifications of an object (device). We designate a series as a set of objects which are designed to be identical. Specific data describing the differing

[1] Even in development efforts, typically, individual test objects remain as members of a series for which specification of the series properties is of use.

[2] For a single level such a hierarchy is usually called a master-detail table pair.

[3] This can be obtained using the 'get_virtual_db' report. See the Database Administrator to run this.

designs are captured in the series_attributes table and series_text_attributes tables which enumerate the values of characteristics described in the series_attributes_defn table. We provide for a hierarchical description of the devices in the series in terms of component devices. This design has been executed such that an arbitrarily deep structuring of object and the components which make it up can be carried out. In terms of magnets, for example, the objects for the series (and subjects) might consist of strand made into cable, cable made into coils, coils assembled into a magnet, and the magnet finally assembled into an accelerator component consisting of two or more magnets. At each step the individual object is well defined with its own serial number and (we hope) its own label (visual_label). But its component parts are traced to the series table with the relationships shown in the series_component (and subject_component) table.

In addition to the newsy[4] information contained in the series table, specific numeric information may be needed about a series or subject. This information can be stored in the series_attributes table possibly duplicating information from columns of the series table but storing it as real numbers (float type) so that comparisons can be made without parsing character data. The specific information on what an attribute is and the units in which it is specified are stored in the series_attributes_defn table. The series_text_attributes table was created specifically so we could store the drawing numbers used to identify parts for inspection. We made it a general table to allow for other, as yet unforeseen, character valued information we may need for a series.

2.1 Description of series Table

The design of the devices should be captured in the series table and the series_components table. Many of the columns in a series table reveal our bias toward testing magnets and are intended to be of a newsy sort to assist both measurement and analysis personnel in making a clear identification of the object from which they have obtained information. We expect to utilize no specific information from the series table during the measurement process except for the series and series_sn and serial_id_width.

[4] We describe as "newsy" the information which is intended not for organizing the project and storing mission critical data but information which is provided to facilitate the browsing of the database by those less familiar with the arcane details of the specific projects. "Newsy" data will allow identification of components and possibly easier comprehension of database relationships.

2.1.1 series

The label for a series should be chosen to have as much mnemonic value as possible. For at least two projects, we have used a pattern in which a magnet series is specified by a three alphabetic characters. The first letter is a specifier for the lamination type, the second for the magnet symmetry type (dipole, quad, ...) and the third letter identifies the magnet length. This is just an example. We allow up to 8 characters (extra length would make general queries *too* bulky and messy).

2.1.2 serial_id_width

In order that applications can check for correct values of the series field, we specify of the permitted field width here.

Usage Parameters

- **Inserts:**LOW
- **Updates:**TEST_MANAGER
- **Deletes:**HOUSEKEEPING
- **Frequency of Access:**SUPPORT
- **Maximum Size:**SMALL

2.2 Description of series_attributes Table

Using design details stored in the series_attributes table, one can determine the nature of a series in detail. One will want to store these details for allowing a user to identify the series and also to allow the TEST_MANAGERS and DEVELOPERS to specify information about the specific designs which can be used by the measurement program for validity checking of input. We assume for now that sufficient information of this sort can be stored using float values.

Usage Parameters

- **Inserts:**LOW
- **Updates:**TEST_MANAGER
- **Deletes:**HOUSEKEEPING

- **Frequency of Access:**SUPPORT
- **Maximum Size:**MEDIUM

2.3 Description of series_attributes_defn Table

The permitted float valued attributes for (all) series are enumerated here. This permits validity checking and also provides a specification for the units of measure.

2.3.1 units

The units for the attribute must be specified here. This forces a consistent set of units across magnet types. Sometimes we 'enjoy' the freedom to choose inches rather than meters. We will miss that here.

Usage Parameters

- **Inserts:**LOW
- **Updates:**DBA
- **Deletes:**HOUSEKEEPING
- **Frequency of Access:**SUPPORT
- **Maximum Size:**SMALL

2.4 Description of series_text_attributes_defn Table

The permitted character valued attributes for series are enumerated here.

Usage Parameters

- **Inserts:**LOW
- **Updates:**DBA
- **Deletes:**HOUSEKEEPING
- **Frequency of Access:**SUPPORT
- **Maximum Size:**SMALL

2.5 Description of series_components Table

The series_component table relates a series with the component of which it is made. Note here the concept of role. An object may by design have several identical components but each of them may have their special role within the object. For example a quadrapole may have four identical coils for some design. They can be placed as coils one, two, three and four. One should therefore identify them as such in their role field.

2.5.1 role

It should be noted that the role of a component in a device (series of devices) should be described in a suitably hierarchical fashion. If intermediate assemblies are physically real but one attempts to bypass them in the database description, the role definition may become convoluted, difficult to understand or imprecise. The role should clearly specify the function of an component in a device.

As an example, consider the Main Injector Dipoles (IDA or IDB). It has two identical half cores. The half cores may be specified as series. Then the specification of the components of the series IDA will require two different uses of the half core: ROLE = TOP HALFCORE and ROLE = BOTTOM HALFCORE.

Usage Parameters

- **Inserts:**LOW
- **Updates:**TEST_MANAGER
- **Deletes:**HOUSEKEEPING
- **Frequency of Access:**OFF_LINE
- **Maximum Size:**SMALL

3 Measurement Subjects

The subjects table parallels the series table. The subjects components table parallels the series components table, but important distinctions must be made. First note that each subject, by our design, must be a member of a series with the series serial number stored in the subject table. Secondly, we need to decide what identifies a subject (physically) so that we can define when it was created or destroyed (or "recreated") and thereby specify when its formal identifier (serial_id) must be reassigned. An object has a unique identity within the framework of a project in which it is to be used, depending on physics and engineering considerations. For example the Tevatron dipoles have traditionally assumed a unique identity based on the collaring process. If one uncollars a set of coils, uses the same coil sets and recollars to a new package, one assigns a new serial number. But completely rebuilding a magnet from the coil on out (new cryostat and yoke) results in the same serial number being applied to the magnet. A different rule could apply in a different project. The above rule seems to make good sense in a superconducting magnet project but little sense for conventional magnets. A useful rule concerning the identity of steel and copper magnets requires that the identity of a magnet is maintained as long as all yoke pieces from an identified magnet remain with the assembly.

A second concept in the subjects table is the concept of rework number. Here we assume that repeat instances of measurements of a given device must reflect whether that device is the same or has been revised (modified) since the previous measurements. The rework number should be incremented each time a modification which makes a physically significant change to a test object has been carried out but which makes no change which causes the serial_id to be changed. The combination of series (SSS), subject (JJJ), rework (RR) displayed as SSSJJJ-RR can be used as a visual identifier for a magnet and provide an easily read, easily understood name for the magnet under test.

Subject component tables allows us to capture in detail any relevant information on the components which actually make up a test subject. The usefulness of this will depend on test or manufacturing lot data which has been captured about given components and any interest in relating that data to the measurement data on the completed test subject. Therefore, not all subjects will have subject components related to them. One or more query steps may be required to make these relationships meaningful. Note that the nature of the relationship (role) between the subject component and the subject is specified via the series_components table.

3.1 Description of subjects Table

The subjects table uniquely identifies a test subject and provides information about its characteristics. Each row specifies a unique test subject. The uniqueness is both in the unique item itself specified by the serial_id and its unique identity in time specified by its rework_id.

3.1.1 Nonstandard Trigger

The nonstandard trigger for this table will ensure that no two entries have the same series, serial_id and rework number (which ensures a unique visual_label for main injector magnets) when the decommissioning_dt is NULL.

3.1.2 serial_id

The serial_id is the identifier which tells which member of a series is the current test subject. We prefer that all serial_id's be specified as numeric characters which can be interpreted as an integer. The number of characters which is nominally utilized is specified by series.serial_id_width. Applications should enforce this requirement on a per-project basis. However, for flexibility we store a varchar(8) (alphabetic or numeric characters) so that special cases can be accommodated.

3.1.3 production_date

We want to uniquely specify the device and identify its character. The production date will help make this possible. However, it is surely not possible to have the production date for all devices, therefore NULL is allowed. We expect to derive the production_date for this serial_id and rework_id from the traveler document which accompanies the test subject but this will be project dependent and should be further specified for each project.

3.1.4 decommissioning_dt

The decommissioning_dt will probably be entered when the magnet is reworked, at the same time the new record for the reworked magnet is entered. It provides us with the ability to ensure, via non-standard triggers, that no two magnets will have the same visual label (the format of which, for main-injector magnets, is given above).

Usage Parameters

- **Inserts:**LOW
- **Updates:**TEST_MANAGER
- **Deletes:**HOUSEKEEPING

- **Frequency of Access:**SUPPORT
- **Maximum Size:**MEDIUM

3.2 Description of subject_components Table

This table is the detail table subsidiary to the subjects table in which the relationship of components to a subject are enumerated. Note that the specific components are identified here but their role is specified by reference to the series_component table.

Usage Parameters

- **Inserts:**LOW
- **Updates:**TEST_MANAGER
- **Deletes:**HOUSEKEEPING
- **Frequency of Access:**OFF_LINE
- **Maximum Size:**MEDIUM

4 Connecting Measurement Subjects to Checklists

4.1 Description of checklists_subject_connection Table

This table is used to specify to measurement systems a relationship between checklists of measurement activities and the actual objects to be tested. Although we have chosen the name checklists_subject_connection for this table, it actually can relate either a subject to a checklist or more commonly a series to a checklist. This is accomplished by allowing the subject_sn field to point to either the subject_sn from the subjects table or to the series_sn from the series table. Note that to enforce this requires a non-standard trigger for that purpose.

4.1.1 subjects_sn

This serial number can be the serial number of a subject or a series so that checklists can be connected with either a specific test subject (magnet) or a series of test subjects.

Usage Parameters

- **Inserts:**LOW
- **Updates:**TEST_MANAGER
- **Deletes:**HOUSEKEEPING
- **Frequency of Access:**SUPPORT
- **Maximum Size:**MEDIUM