The Use of Databases in Telemetry Processing Systems

THE USE OF DATABASES IN TELEMETRY PROCESSING SYSTEMS

James E. Massing
Senior Principal Engineer
Fairchild Weston Systems, Inc.
(Formerly EMR Telemetry)
P. 0. Box 3041
Sarasota, FL 34230-3041

ABSTRACT

The amount of information necessary for a computer to setup and process data through a telemetry system is increasing rapidly, The complexity and number of telemetry formats often requires a variety of front-end setups to be kept in the computer. Also, as hardware preprocessors become more common, there is a need to maintain in the computer all information used to direct real-time processing of individual parameters. The need for storage of those setups and parameter definitions, and rapid retrieval of this information, has led to the use of databases in well-designed telemetry systems.

INTRODUCTION

One requirement for proper use of databases is the optimum division of information into separate modules. This permits modules of one system database to change and not result in modification of the data contained in other modules of the database. Another requirement, and possibly the most difficult problem to solve, is to create a man/machine interface capable of presenting database setup information to the operator in a usable format. This interface must give the user the ability to configure all resources in the system while not being overpowered with endless options.

Experience has shown that division of the information in three separate areas is best. These three databases are;

- Parameter related or format internal definitions
- Stream related of format external definitions
- Telemetry system configuration.

This paper will focus on the first of these three databases, the parameter database.

PARAMETER DATABASE

Definition

The parameter database provides a standard location and format for the storage of information specific to each attribute being measured by the telemetry system. The items being measured, such as vibration, or velocity, are referred to as “parameters”. The database, while it does not contain actual measured values for these parameters, contains the information which other system software needs in order to store actual values and convert them for display and analysis. Parameters received in all streams of the system may be defined in one database. Alternately, separate database files can be maintained to distinguish a telemetry stream (PCM, FM, PAM), or group of streams, from each other.

Storage Format

The parameter database is a set of three types of files containing fixed length records. The first file type is a key file. These files contain primary parameter file record numbers grouped by the various key values that have been used. The second file type is the primary parameter data file. The database primary file contains information which is required for all parameters being defined. This includes parameter name, parameter type, position and length in the telemetry format, engineering unit name, etc. The third file is the secondary parameter data file. The database secondary file contains real-time data processing information in the form of algorithm sequences and arguments. There can be multiple sets of secondary records, each set define a processing path. A processing path is a sequence of algorithms and algorithm arguments that define the operations needed to support any processed output (i.e. archived data, display data or DAC/strip chart data).

The primary and secondary file construct is used to utilize file space efficiently since the amount of data used to define a parameter can vary greatly. The database storage format is based on this series of files containing fixed length records. The key files have very short record structures thereby allowing for very rapid access and small storage size. The primary and secondary files contain larger records but they are still fixed length. In the primary file there is one record per parameter, while the secondary file contains as many records as necessary to contain all of the required information.

Information typically stored in the parameter database includes parameter name, parameter type, stream type, engineering unit conversion constants, and preprocessing algorithm specifications. The data is organized by a predefined format, called a schema. Refer to Figure 1 for the representation of the interaction of these three file types.

PARAMETER DATABASE ACCESS

User applications will extract all the information needed about a particular parameter from the parameter database. The particular piece of information needed will define whether the application has to access the secondary data or if the information at the primary level will be sufficient. Typically, host computer based operations, data display and report generation, will need only primary information. Secondary information is needed for setup and configuration of the preprocessing hardware and real-time software.

All user applications must start their database access with a call to an initialization routine. This will open the necessary files and setup data structures common to the entire system and private to the user application. The first call the application makes for data will be a parameter access routine. The application will supply as many of the key values as needed to uniquely identify the requested parameter or group of parameters. Each record in the database contains key fields. The following are examples of possible key fields;

- Stream Name (PCM, FM, PAM, etc.)
- Parameter Name (16 characters alphanumeric)
- Parameter Type (analog, digital, discrete, time)

Each record in the key files contain key field values of a particular parameter and the record number in the database primary file where that parameter's primary information is stored. Indexing into the key files is done by using a hashing function. A hashing function uses some attribute of the key and produces a numerical value. This value is record number of an entry in the key file. This record number defines the beginning point and limits the bound of the linear search for an exact match on the user specified key value. The primary record file number associated with the matched keys is read from the primary file and its contents returned to the calling application. The information returned includes parameter name and type, stream name, engineering unit name plus a series of pointers into the secondary file.

At this point the application has three different options for continuing database accesses. The first is a new parameter access call. This will extract an entirely new parameter's primary record information based on the supplied key field values. The second option is a next parameter access call. Assuming that not all of the available key field's were specified in the original access call, a next parameter call will return the primary record information associated with the next parameter in the database that has the same key field values. If the application had originally specified only a stream name, the returned parameter would be the next parameter in that stream definition. If the application had originally specified a stream name and a parameter type, the returned parameter would be the next parameter of the specified type in that stream definition. The lost option is a

database access call that would produce a detailed processing definition for one of several processing paths that parameter will take through the telemetry preprocessor. This sequence of operations is shown in Figure 2.

PARAMETER DATABASE EDITOR

Menu Standards

In any man machine interface, three items must be present. First, the information must be displayed in a concise and orderly fashion. Second, entries must be validated to the extent possible when entered. Third, and possibly the most important, no piece of information should be requested in more than one place and never defined in more than one way.

A data entry protocol needs to be established for the menus. An example of this protocol is described in the following statements.

- The currently selected entry will be highlighted.
- The type of input required will be shown by a prompt on the bottom line of the display.
- Operator input will be echoed following the prompt until the <CR> is entered.
- Invalid input will produce an error message on the bottom line of the display. The message will remain until acknowledged by pressing the <CR>.
- The “DELETE” key will backspace to permit the correction of typing errors.
- The “ARROW” keys will select the next entry in the indicated direction.
- Vertical “ARROW” key movement “wraps-around” between the top entry and the bottom entry.
- Horizontal “ARROW” key movement “wraps-around” and also moves up or down to the next row of entries.
- The “RIGHT-ARROW” moves down the screen when it “wraps-around” while the “LEFT-ARROW” moves up.
- The “CONTROL” keys may direct system actions as indicated below.

    - P - Print screen
    - U - Cancel current operation
    - D - Delete current parameter
    - R - Return to previous menu
    - Z - Terminate current operation

Man/Machine interface

The operator communications are accomplished via interactive menus which run on a terminal or work station. Examples of these menus are shown in Figures 3 through 5. Options are provided to create, edit or delete parameter definitions. It should be noted that once a parameter has been edited and saved that the previous definition data is gone (overwritten by the new definition data). In addition, due to the possible combinations that the telemetry frontend equipment can be adapted to, the editor may not be able to perform adequate cross checking on the parameter definitions. It is the users responsibility to ensure the correlation of all the parameters with themselves and with the intended system configuration that the defined stream is to be run on.

The database editor uses operator input and the menus as the principal means by which a user may interact with the parameter database system. Standard menu techniques are used to manipulate the screens, with the addition of a virtual page feature. This virtual page feature will allow inspection and editing of data sets which are too large to fit on a screen all at once.

The virtual page menu screen will contain a “window” through which a fixed number of lines of parameter information may be inspected or edited. The operator will be able to scroll up or down through the lines, by either one line at a time or a specified number of lines in either direction. This type of display will allow the user to view or edit large amounts of parameter information on one menu screen, instead of on several different screens, thus avoiding any confusion that might be associated with using multiple screens to display parameter information which is related.

Error messages are displayed at the bottom of the screen in the prompt field. These messages must be acknowledged by a carriage return before editing may continue. The error messages are also preceded by a mnemonic which denotes the source and number of the error. For instance, DB09 might represent an over limit validity check within the parameter database editor.

An editor is used to create and maintain the parameter database. The user starts this editing or creation processing by specifying the key file values for the desired parameter. If a parameter with these keys exist the session is assumed to be an edit session. If there is no direct match of the key values, a new parameter is created with the supplied key values. This is accomplished by using the menu screen shown in Figure 3. The operator acknowledges that the correct values have been entered by entering a “GO” command (CTRL G). This sends him to the parameter identification, output and process selection screen shown in Figure 4.

The parameter identification, output and process selection screen is split into four separate sections. The top section repeats the parameter identification information entered on the previous screen. The bottom section is set aside for entering miscellaneous information. The center left section defines the source of the data. The center right section defines the uses of the data.

The miscellaneous information is normally information needed by the host computer for the purposes of data display and report generation. The section defining the source of the data will change based on the system configuration. If the source of the data is a frame synchronizer, a data tagger or a MIL-1553 bus the prompts will change and request different pieces of Information. The uses of the data are again dependent on the system configuration. Typical data usages are archive storage, data displays and DAC/strip charts. These all have different data processing requirements and require different processing algorithm sequences. All information to this point has been at the key file and primary database file level.

Taking the scenario one step further is show in Figure 5 the processing path setup screen. This screen shows the parameter identification information entered on the fist screen as well as the sequence of algorithms that will transform the input data to the desired output format. Definition of the arguments for each algorithm makes the execution of that algorithm unique for the this parameter.

CONCLUSION

During the 1990's, the use of databases in telemetry systems will become common place. There will be two areas that will become of great interest to the managers and users of these database oriented systems. First, the ability of the system to accept and maintain the database information. Second, the ability to transfer the database information from one vendors equipment to another.

The method for presenting the information will be selected without regard for the underlying system architecture. Many operators will want a mouse and a graphical interface while many others will prefer arrow keys and text entry. Man machine interfaces currently being utilized in the personal computer market will soon become a part of the telemetry community. Operator interfaces will vary between vendors and selection will be made by personal preference.

The transfer of information between different vendors equipment will be a much more difficult problem to solve. It will require the participation of a governing body such as ISA or the IRIG.

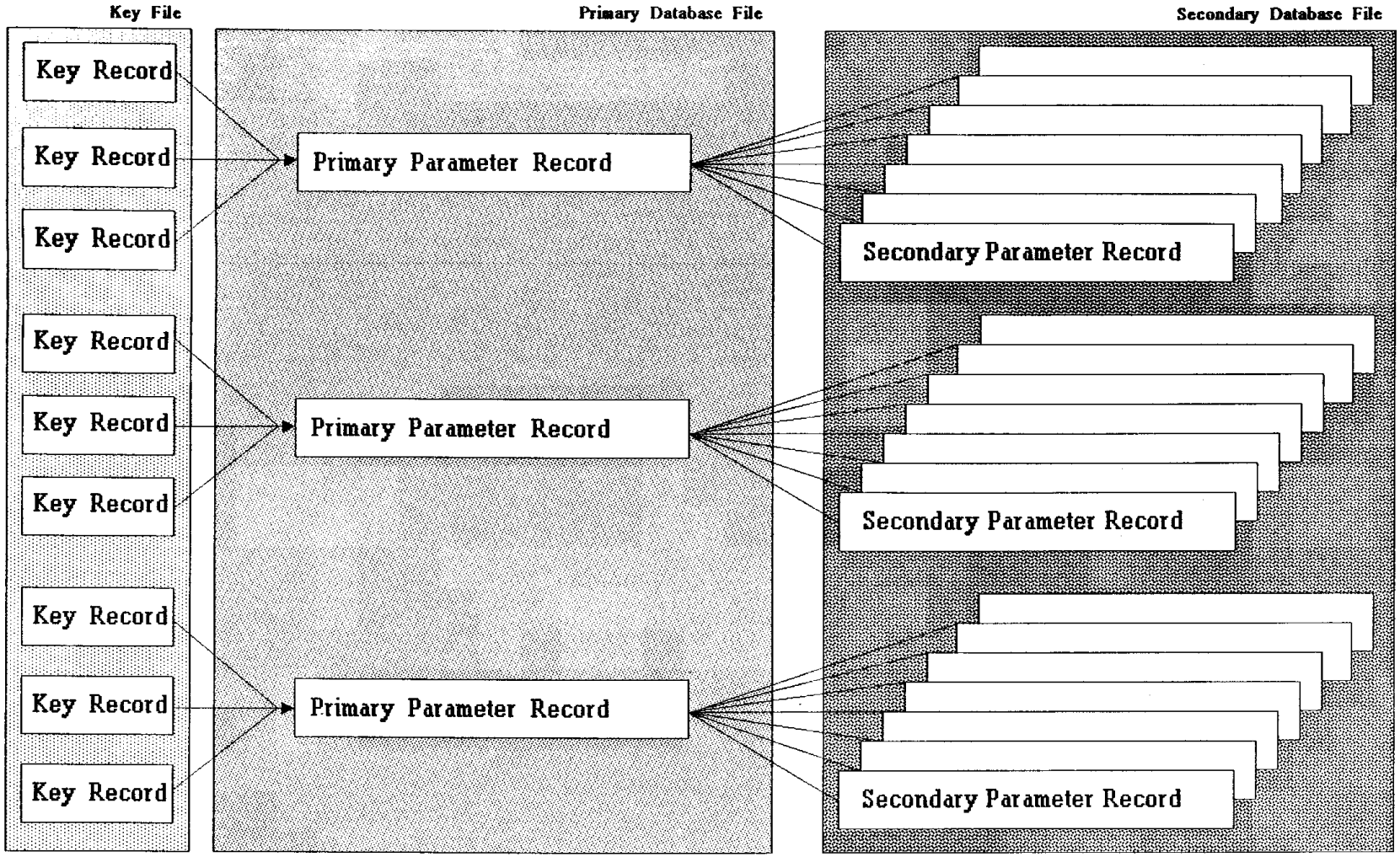

Figure 1: Parameter Database File Structure

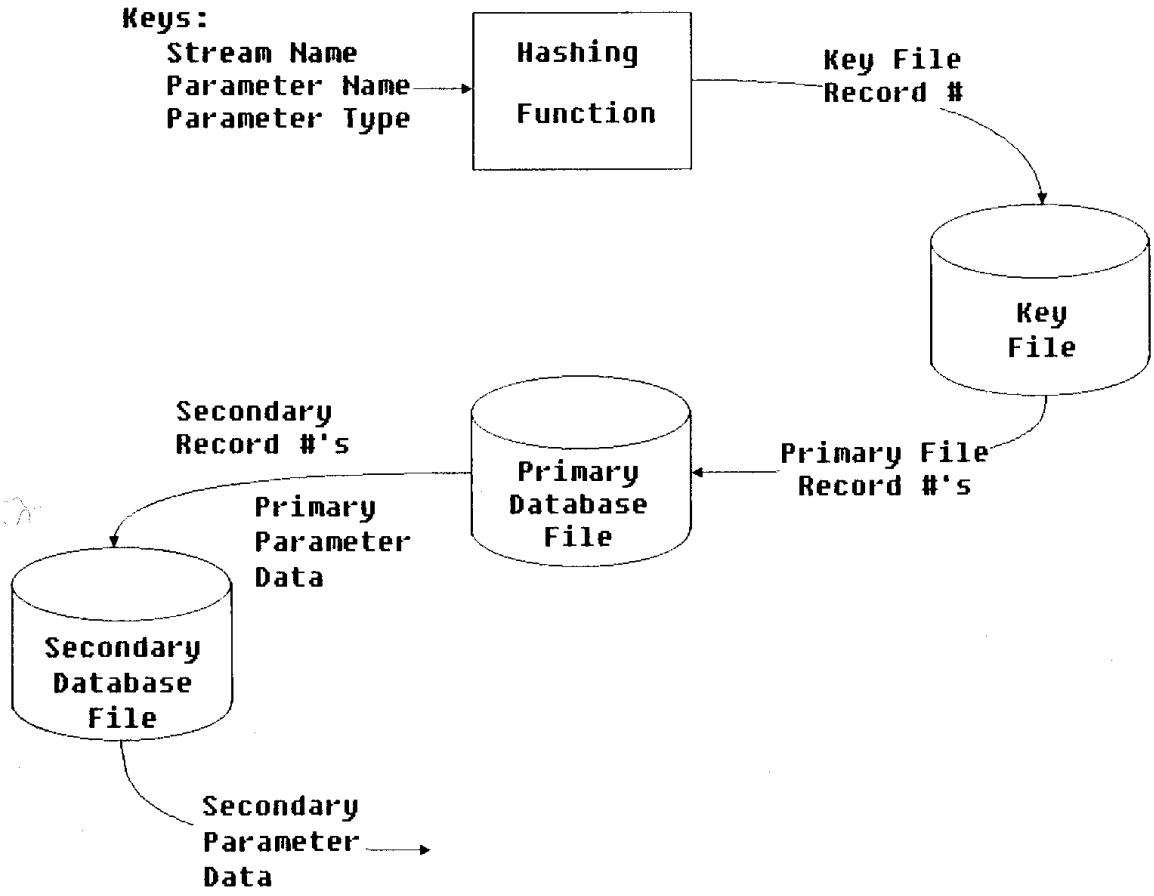

Figure 2: Parameter Database File Access

PARAMETER DATABASE EDITOR (dd-mon-yr) (hh:mm:ss)

STREAM NAME = [redacted]

PARAMETER NAME =

PARAMETER TYPE =

Enter CTRL-H when you need instructions

Dynamic updating is ENABLED

CTRL-keys: D=Delete, G=Go, H=Help, P=Print, W=Refresh, Z=Terminate

CHARACTER:

Figure 3: Parameter Selection Screen

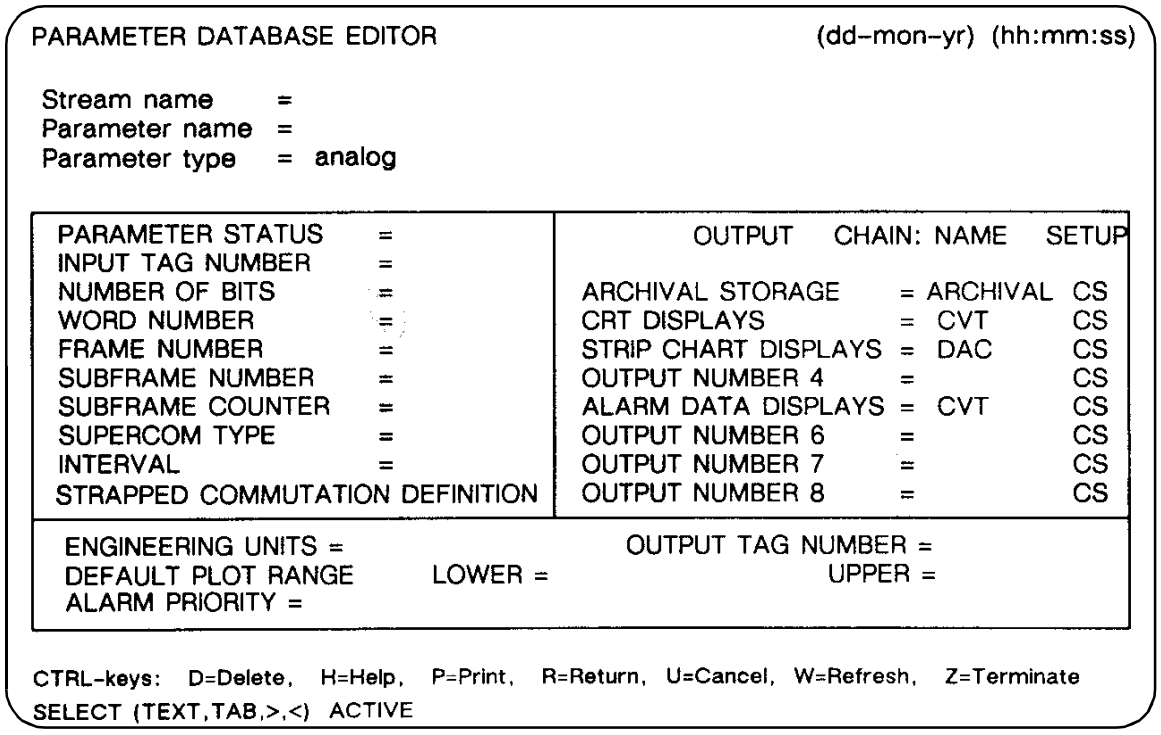

Figure 4. Parameter Identification, Output and Process Selection Screen

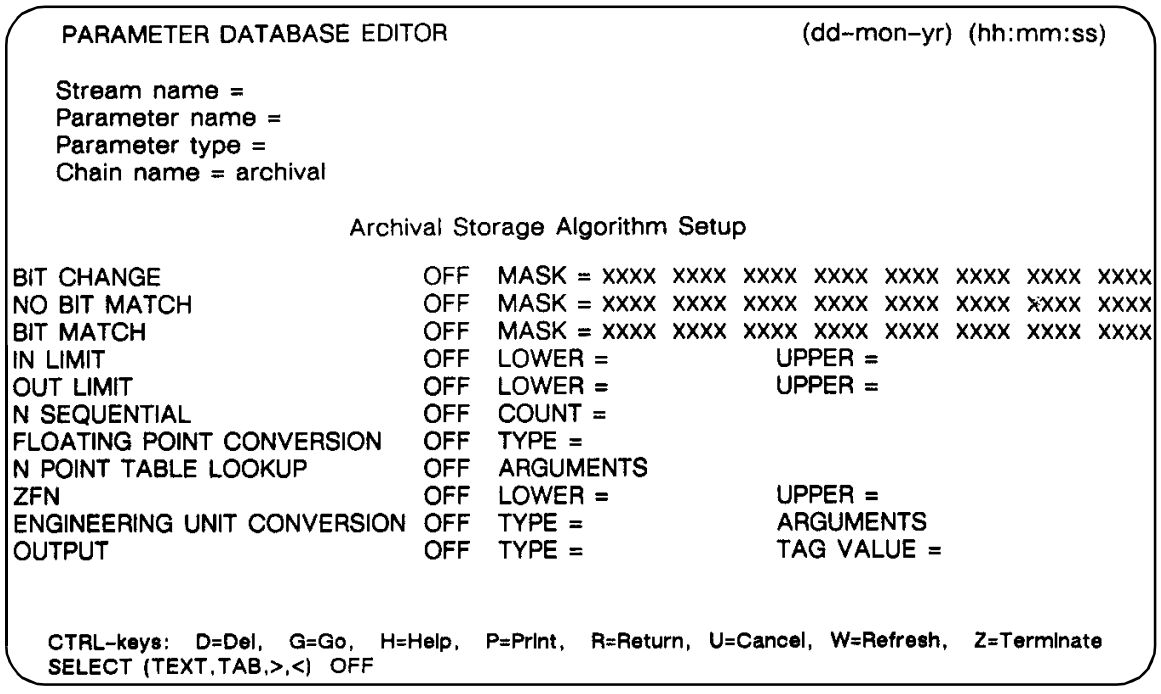

Figure 5: Processing Path Setup Screen