MADAM: An Example of Hierarchical Data Set Management

Richard L. Gimarc, Boole & Babbage, Inc.

1. Abstract

The Model 204 Accounting Data Management System (*MADAM*) is a system designed to extract, summarize, manage, and report user–accounting data produced by the Model 204 database management system.

What makes *MADAM* unique from other data collection and management systems built with SAS is its novel technique for managing its collection of summary data sets.

MADAM maintains a 2–level hierarchy of summary data where each level contains data collected at a different degree of granularity. Intelligence is built into the system which enables *MADAM* to merge data collected at level 0, daily data, to produce level 1 monthly summary data. The only user interaction required is to identify the target month. The user does not need to know the names of the daily data sets used in the monthly merge.

The technique presented for managing a two–level hierarchy of SAS data sets can easily be extended to manage an n–level hierarchy. Application of this technique would prove valuable in the area of capacity planning, for example, where the planning horizon dictates the granularity of data used; weekly, monthly, quarterly, or yearly.

2. Background

Model 204 is a database management system developed, maintained, and marketed by Computer Corporation of America and is designed to address the database needs of large data centers. As with most database systems, Model 204 maintains a variety of utilization statistics for itself and its users. These utilization statistics along with other data describing a Model 204 run are logged on the Model 204 Journal data set. The Journal is maintained as a file of variable length records. Each record consists of a record header, one or more Journal entries, and a record trailer. There are a variety of Journal entries identified by a type field.

One entry type of particular interest is the type 9 user logout statistics Journal entry. The information contained in this entry describes the resources consumed by a particular user and is necessary for apportioning the cost involved with running Model 204.

3. Design Goals

The primary design goal of *MADAM* was to develop a system to extract and report user activity in Model 204. This data was necessary for billing and monitoring Model 204 usage. With this as a base, the following design goals were established:

1. Develop a system to extract user–accounting data from the Model 204 Journal data set.
2. Create a database of user–accounting data suitable for billing and monitoring Model 204 usage.
3. Minimize user interaction. The system must be as automated as possible.
4. Provide for recovery when failures occur.

The system developed addressed these four goals and is described in the following sections.

4. MADAM Components

Before describing the operation of *MADAM*, the components which make up the system will be presented.

4.1 MADAM Data Sets

The *MADAM* system is based on three data sets. Two of the data sets, the *Daily* and *Monthly* data sets, are SAS data bases. The third data set is an OS generation data set and is refered to as the *History* data set.

Daily Data Base

The *Daily* SAS data base contains a collection of *Daily* SAS data sets, one for each day of *MADAM* collected Journal data. The *Daily* data sets are named

D*yymmddd*

where *yy* is the year, *mm* is the month and *ddd* is

the Julian date. Each observation of a *Daily* data set contains the accumulated Model 204 usage data for an individual user during day *ddd* of month *mm* and year *yy*.

Monthly Data Base

Similarly, the *Monthly* SAS data base contains a collection of *Monthly* SAS data sets, one for each month of *MADAM* collected Journal data. The *Monthly* data sets are named

M*yymm*

where *yy* is the year and *mm* is the month. Each observation of a *Monthly* data set contains the accumulated Model 204 usage data for an individual user during month *mm* of year *yy*.

History Data Set

The *History* data set serves as a directory of the current collection of *Daily* data sets. A new generation is created whenever *Daily* data sets are created or deleted. Each record of the *History* data set contains the name of a *Daily* data set.

4.2 MADAM Programs

There are two programs which are used to drive *MADAM*. Journal_Reader is the SAS program which reads the Model 204 Journal data set. The output of this program is a single SAS data set containing a summarization of the type 9 user–accounting Journal entries read. The second program is a PL/1 program named SAS_Generator. This program generates SAS programs where the type of program generated is specified through input parameters passed to the program.

4.3 MADAM Procedures

The facilities of the *MADAM* system are invoked with the following four JCL procedures:

BUILD_DAILY

The BUILD_DAILY procedure reads the Model 204 Journal data set and builds the corresponding *Daily* data sets (see Figure 1).

The first step of this procedure is the SAS program Journal_Reader which reads the Model 204 Journal data set. Journal_Reader reads the Journal data set and selects the type 9 user–accounting entries for further processing. The output of this program is a single SAS data set containing all selected Journal entries and a temporary *dates* OS data set which contains one date record per day detected on the Journal file.

The program SAS_Generator is called in step 2 of this procedure. Parameters are passed to SAS_Generator which direct it to generate a SAS program to read the SAS data set created in step 1 and partition it into the corresponding *Daily* data sets. SAS_Generator reads the *dates* file written in step 1 to determine the names of the *Daily* data sets to be created. Since new *Daily* data sets are being created, the *History* data set is updated (create a new generation) to contain the names of the new *Daily* data sets.

Step 3 executes the SAS program generated by SAS_Generator in step 2. This program reads the SAS data set created by step 1 and partitions it into the corresponding *Daily* data sets.

The end result of executing this procedure is a set of *Daily* data sets containing the accumulated Model 204 usage data for all users whose Model 204 sessions were logged on the Journal data set.

BUILD_MONTHLY

This procedure collects the *Daily* data sets for a given month and creates the corresponding *Monthly* data set (see Figure 2).

Step 1 is a call to SAS_Generator with parameters directing it to generate a SAS program to merge the *Daily* data sets for a given month into a single *Monthly* data set. SAS_Generator reads the *History* data set to determine the names of the *Daily* data sets to be merged for the selected month.

Step 2 executes the SAS program generated in step 1 to build the *Monthly* data set for the selected month.

BUILD_MONTHLY_&_DELETE_DAILY

This procedure is identical to the BUILD_MONTHLY procedure except that the *Daily* data sets used to build the selected *Monthly* data set are deleted. In addition to deleting the *Daily* data sets, the *History* data set is updated to exclude the *Daily* data sets deleted after the *Monthly* merge.

GENERATE_BILL

GENERATE_BILL is used to generate custom billing records for a given month.

Step 1 is a call to SAS_Generator with parameters identifying the month and the type of program required. The output of this step is a SAS program which reads the corresponding *Monthly* data set and writes custom billing records.

The SAS program generated in step 1 is executed in step 2. The output of this step is a sequential data set containing billing records describing Model 204 usage for each user during the selected month.

5. Using MADAM

MADAM is currently being used by the Texas Education Agency (TEA) in Austin, Texas to collect, manage, and report Model 204 user–accounting data. TEA's use of the system will now be described.

Model 204 is used daily at TEA. Journal data is written to a generation data set so that a new generation exists for each Model 204 run. Approximately every week the BUILD_DAILY procedure is run the extract user–accounting data from that week's Journal data sets and build the corresponding *Daily* data sets. At the end of each month the BUILD_MONTHLY_&_DELETE_DAILY procedure is run to build the month's *Monthly* data set. GENERATE_BILL is then run to generate input to TEA's computer billing system.

Three points are worth noting. First, as *Daily* data sets age they are removed from the system. Most processing of Model 204 user–accounting data at TEA uses *Monthly* data. The *Daily* data sets serve as a staging point between raw Journal data and *Monthly* data.

Second, user interaction is minimal. All parameter input required by the *MADAM* procedures are expressed in terms the users is familar with. To build *Daily* data sets the user only needs to identify what Journal data sets to use as input. The naming of the *Daily* data sets that are built is automatic and standardized. End–of–month processing requires the user to identify the month and year. No knowledge of what *Daily* data sets exist or of the naming of the *Monthly* data set being built is required.

And third, recovery from failure is built into the procedures used to run *MADAM*. With a rigorous disk backup procedure in place, the *MADAM* system can easily be recovered and restored to a consistent state following a failure.

6. Applications

The technique used by *MADAM* for managing its 2–level hierarchy of SAS data sets can easily be extended to manage an n–level hierarchy where level i contains an aggregation and summarization of level i–1 data.

Level 0 is the level at which data enters the system. This level consists of a collection of level 0 SAS data sets, each containing data collected at the same degree of granularity. A *History_0* data set will be maintained to serve as a directory of the level 0 data sets.

To step from level 0 to level 1, a rule must be formulated which identifies which level 0 data sets map into a level 1 data set. Creating level 1 data sets involves applying the rule and using the *History_0* file to identify the level 0 data sets to be used. When a new level 1 data set is created it will be logged in the level 1 directory data set, *History_1*.

The generalization follows. To implement a hierarchical data set management scheme such as this two design decisions must be made:

1. Level i data sets naming conventions.

2. Rules for identifying which level i data sets to merge to create a level i+1 data set.

For *MADAM* this was easy since dates were encoded in the *Daily* data set names.

7. Summary

This paper has presented a system for managing a hierarchy of SAS summay data sets where each level of the hierarchy contains data collected at a different degree of granularity.

It has been demonstrated that the user interface for managing such a hierarchy can be simple. No detailed knowledge of the names of individual data sets at each level of this hierarchy is necessary. A companion hierarchy of directory data sets can be used to automate the staging of data up the hierarchy.

MADAM is an example of hierarchical data set management. The technique used by *MADAM* can easily be extended to other application areas where data is kept at various degrees of granularity.

8. Author Contact

Boole & Babbage, Inc.
611 West 14th Street
Austin, Texas 78701

(512) 478–0788

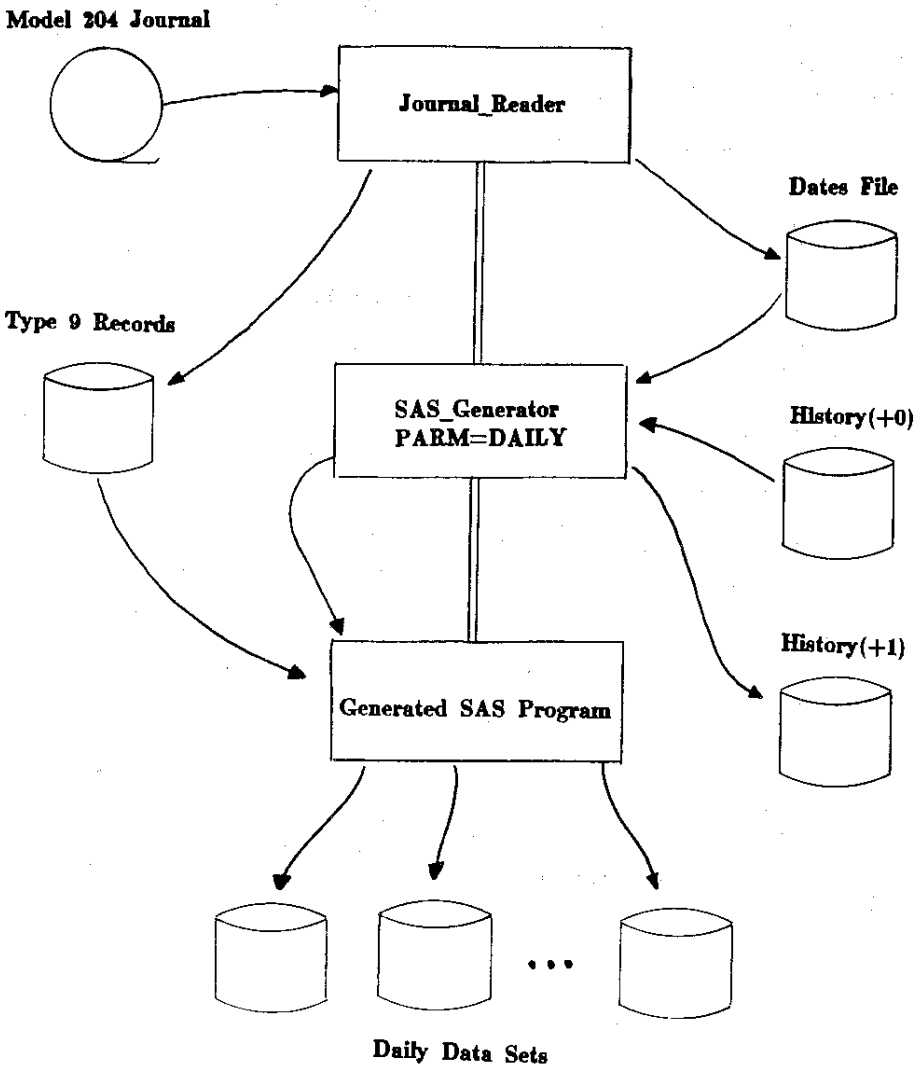

Figure 1. BUILD_DAILY Procedure

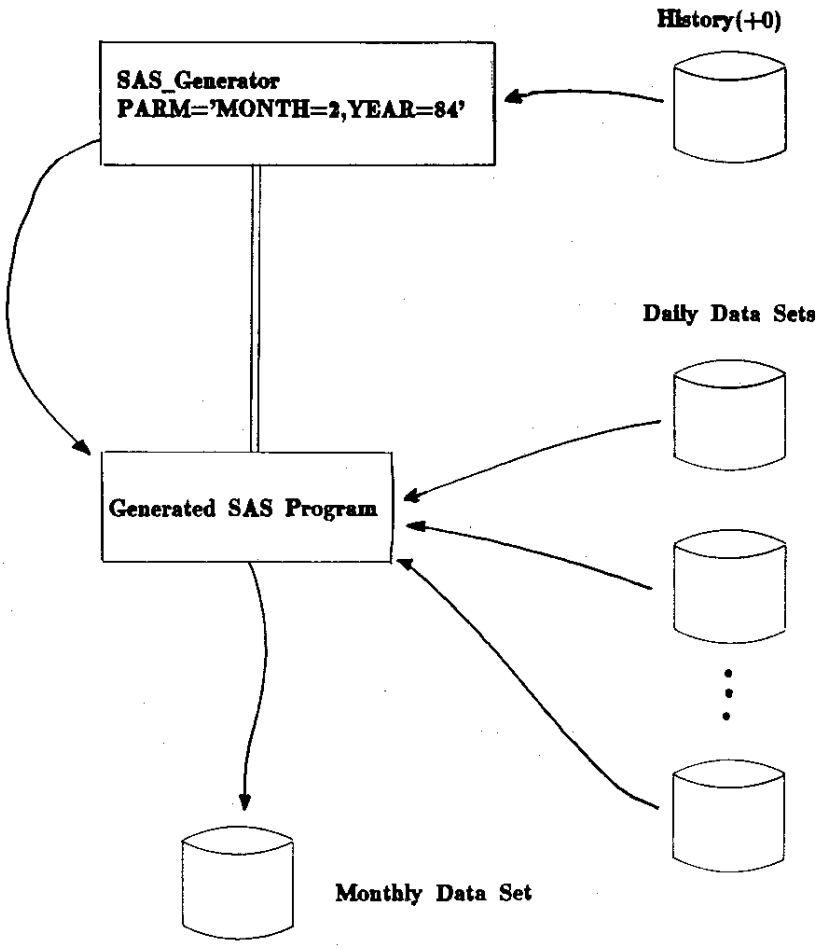

Figure 2. BUILD_MONTHLY Procedure