Two information systems have been developed at the U.S. Department of Agriculture, Soil Conservation Service (SCS) using base SAS software. The two systems are the Equipment Management Information System (EMIS) and the Real Property Management Information System (RMIS). The EMIS system stores information about vehicles owned and leased by SCS. This information includes vehicle utilization data, operation data, and maintenance data. The RMIS system contains information about real property owned and leased by SCS. It is a centralized source of the real property holdings of the agency. The information stored in it affords the ability to track space utilization; identify under-utilized property; and provide a reference for answering inquiries from Congress and others.

This paper will discuss the structure of the two systems and SAS language tools used in reporting from and updating the systems.

The Equipment Management Information System (EMIS) and the Real Property Management Information System (RMIS) are two systems at the USDA Soil Conservation Service developed using base SAS software. EMIS is a tool for managing the fleet of vehicles owned and leased by SCS. RMIS is a tool for managing real property owned and leased by the agency. The systems are comprised of SAS data sets together with SAS language programs for reporting and updating.

The two systems reside on a NAS 9080 (IBM plug compatible) mainframe in Washington, DC. The systems are accessed from remote sites in a batch mode and at the central site both interactively and in batch mode. The remote sites are the over fifty SCS state and regional offices. In these locations Harris 1650 minicomputers are used to create Job Control Language (JCL) streams to be transmitted to the batch queue of the mainframe.

Data Entry and Verification

Data entry programs written in Regal, an interactive language for the Harris 1650 minicomputer, are used to provide transactions to update the EMIS and RMIS systems. The data entry program for EMIS provides eight screens. Property management officers or their designees enter vehicle information into the screen formats. Some data are rejected at the screen, and the user cannot proceed until the data are corrected. The data entry program creates nine records from the screens. The records comprise a raw data set to be processed by SAS to create a temporary SAS data set of transactions to update the data base.

The RMIS data entry program works similarly. In this system five screens are provided, and fifteen records are created from each set of screens.

The Data Bases

The EMIS data base is a SAS data set on disk file format. It is a flat file structure with 169 variables and about 12,000 observations. The updating of it is complicated by the maintenance of quantitative data. The end user, for example, enters monthly miles driven, quarts of oil used in a month, and gallons of fuel for a month. The update program computes and stores the data in the data base.

The EMIS system provides both batch and on-line reports. At the remote sites, batch reports are invoked via catalogued procedures. At the central site batch reports are invoked via TSO CLIST's which submit jobs to the batch job queue. On-line reports are implemented via TSOCLIST's which allocate necessary data sets and establish the processing environment. Then interactive SAS is entered, and prewritten SAS programs are invoked via the %INCLUDE statement.

One report of particular interest is the Vehicle Age and Mileage Analysis. (See figure 1.) This report requires a running tally of vehicles by type (sedan, station wagon, etc.), year/model, and mileage range. There are nine vehicle types, eleven year/models (plus 1 total), and ten mileage ranges (plus a total). A three-dimensional array - 9X12X11 - was defined to facilitate the processing. The SAS implementation of this data structure required the definition of 118 arrays. In SAS the arrays must be defined bottom up; that is, the arrays which are elements of arrays must be defined before the arrays containing them. This required 108 array statements each defining eleven mileage ranges; nine array statements each defining twelve year/models; and one array statement defining nine vehicle classes. The 108 arrays are defined 12 at a time followed by one array containing year/models. Nine sequences of these are defined followed by the one array definition for the vehicle classes. (See figure 2.)
Once the arrays have been defined, the algorithm becomes rather trivial. For each observation (a vehicle):

- set \( I \) from 1 to 10 based on the mileage of the vehicle;
- set \( J \) from 1 to 11 based on the vehicle year/model;
- set \( K \) from 1 to 9 based on the vehicle type.

Once \( I, J, \) and \( K \) have been set, tally in the appropriate cell, \( (\text{CLASSV}=\text{CLASSV}+1) \). To tally in the total by mileage, \( I \) is set to 11 while \( J \) and \( K \) remain constant. To tally in the total by year/model, \( J \) is set to 12 while \( I \) and \( K \) remain constant. Finally, to tally in the total by mileage and year/model, \( I \) is set to 11, \( J \) is set to 12, and \( K \) is held constant. On end of file one report page is generated for each vehicle type. This demonstrates that careful planning of data structures simplifies programming.

A Useful Application of the Update Statement

The RMIS data entry program creates update transactions to add new installation observations, to change existing installation observations, and to delete existing installation observations. The transactions are part of a transaction data set to be used in an UPDATE statement. However, it is not desirable to add a record that already exists in the master file. Nor is it desirable to attempt to change or delete a record that does not exist in the master file.

In the RMIS system the transactions to be applied to the master file are validated before they are applied. The validation assures that the conditions listed above are not present. The validation is accomplished using the UPDATE statement. A SAS data set containing a list of valid identifiers is derived from the data base. There is one occurrence in the list for each occurrence in the data base. The list of valid identifiers is compared to the identifiers in the transaction data set. The approach will cause the number of transactions in the transaction data set to be reduced or remain the same. The upper portion of figure 3 contains a list of valid identifiers; the lower portion contains a list of transactions to be applied to the data base.

Figure 4 contains the SAS statements used to perform the validation.

1) The DATA statement creates a new SAS data set containing validated transactions (INSTTRAN).
2) In the Update statement:
   a) The list of valid installations (VALINST) is used as the master data set. The IN= variable is VIN.
   b) The transaction data set is INSTTRAN; the IN= variable is INS.
3) The master data set contains only those variables on the BY statement.
4) When the master data set contributes to the observation and the transaction data set does not, the observation is deleted. This prevents the creation of new transactions.
5) When both the master and transaction data sets contribute to an observation and the action code is 'A' meaning add, the observation is deleted. This condition indicates that the record to be added already exists in the data base. The deletion prevents the insertion of duplicate records.
6) When the master data set does not contribute to an observation, and the transaction data set contributes, and the action code is 'C' meaning change 'D' meaning delete, the observation is deleted. This prevents the attempt to change or delete a record that is not in the data base.

The result is a set of validated transactions ready to be used to update the RMIS data base.
### Annual Motor Vehicle Age and Mileage Analysis

**Fiscal Year Ending September 30, 1985**

<table>
<thead>
<tr>
<th>Soil Conservation Service</th>
<th>Total Miles Operated 2,404,712</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vehicle Class - Station Wagons</strong></td>
<td></td>
</tr>
<tr>
<td>Current Year 1985</td>
<td>5</td>
</tr>
<tr>
<td>Preceding Year 1984</td>
<td>4</td>
</tr>
<tr>
<td>3rd Preceding Year 1983</td>
<td>1</td>
</tr>
<tr>
<td>4th Preceding Year 1982</td>
<td>1</td>
</tr>
<tr>
<td>5th Preceding Year 1981</td>
<td>2</td>
</tr>
<tr>
<td>6th Preceding Year 1980</td>
<td>2</td>
</tr>
<tr>
<td>7th Preceding Year 1979</td>
<td>1</td>
</tr>
<tr>
<td>8th Preceding Year 1978</td>
<td>1</td>
</tr>
<tr>
<td>9th Preceding Year 1977</td>
<td>5</td>
</tr>
<tr>
<td>10th Preceding Year &amp; Older</td>
<td>5</td>
</tr>
</tbody>
</table>

**Total:**

<table>
<thead>
<tr>
<th>Model</th>
<th>NUMBER OF VEHICLES BY CUMULATIVE MILES OPERATED</th>
</tr>
</thead>
<tbody>
<tr>
<td>999</td>
<td>1</td>
</tr>
<tr>
<td>9999</td>
<td>6</td>
</tr>
<tr>
<td>99999</td>
<td>5</td>
</tr>
<tr>
<td>99999</td>
<td>1</td>
</tr>
<tr>
<td>99999</td>
<td>5</td>
</tr>
<tr>
<td>99999</td>
<td>4</td>
</tr>
<tr>
<td>99999</td>
<td>3</td>
</tr>
<tr>
<td>99999</td>
<td>4</td>
</tr>
<tr>
<td>99999</td>
<td>2</td>
</tr>
<tr>
<td>99999</td>
<td>1</td>
</tr>
<tr>
<td>99999</td>
<td>4</td>
</tr>
<tr>
<td>99999</td>
<td>7</td>
</tr>
<tr>
<td>99999</td>
<td>10</td>
</tr>
</tbody>
</table>

**Fleet on Hand Oct. 1, 1984:** 50

**Vehicle Years Operation:** 49.1

---

### Data Helix

```
DATA NULL;
ARRAY IPER1(I) Al-I11;
ARRAY IPER2(I) BI-I11;
ARRAY IPER3(I) CI-I11;
ARRAY IPER4(I) DI-I11;
ARRAY IPER5(I) EI-I11;
ARRAY IPER6(I) FI-I11;
ARRAY IPER7(I) GI-I11;
ARRAY IPER8(I) HI-I11;
ARRAY IPER9(I) JI-I11;
ARRAY IPER10(I) KI-I11;
ARRAY IPER11(I) LI-I11;
ARRAY IT1(I) TP1(I) IVER1-IVER12;
ARRAY IT2(I) TP2(I) AAI-AA11;
ARRAY IT3(I) TP3(I) BBBI-BBBB11;
ARRAY IT4(I) TP4(I) CCCI-CCCI1;
ARRAY IT5(I) TP5(I) DDDI-DDDI11;
ARRAY IT6(I) TP6(I) EEEEI-EEEEII;
ARRAY IT7(I) TP7(I) FFFI-FFFFII;
ARRAY IT8(I) TP8(I) GGGGI-GGGGII;
ARRAY IT9(I) TP9(I) HHHHI-HHHHII;
ARRAY IT10(I) TP10(I) II1-II11;
ARRAY IT11(I) TP11(I) III1-III11;
ARRAY IT12(I) TP12(I) JII1-JII11;
ARRAY IT13(I) TP13(I) IIIII-IIIIII;
ARRAY IT14(I) TP14(I) AIII-AIII11;
ARRAY IT15(I) TP15(I) BBIII-BBBIII11;
ARRAY IT16(I) TP16(I) CCCI1-CCCI1;
ARRAY IT17(I) TP17(I) DDDI1-DDDI11;
ARRAY IT18(I) TP18(I) EEEE1-EEEEII;
ARRAY IT19(I) TP19(I) FFFI1-FFFFII;
ARRAY IT20(I) TP20(I) GGGG1-GGGGII;
ARRAY IT21(I) TP21(I) HHHH1-HHHHII;
ARRAY IT22(I) TP22(I) IIIIIII-IIIIII11;
ARRAY IT23(I) TP23(I) JIIIIII-JIIIIII11;
ARRAY IT24(I) TP24(I) IIIIIIIII-IIIIIIII11;
```

---

### Figure 2

```c
334
```
(1) DATA INSTRN;
(2) UPDATE RMIS. VALINST(IN=VIN) INSTTRAN(IN=INS);
(4) IF INS=O
(6) DELETE;
END;
END;

THEN DO;
PUT $0 "THE FOLLOWING INSTALLATION CANNOT BE ADDED"
"TO THE FILE, IT IS CURRENTLY EXIST IN THE FILE:";
900 "STATE" 931 "STATE"
837 "CITY" 843 "CITY"
850 "COUNTY" 857 "COUNTY"
861 "INSTALLATION NUMBER" 861 INST_NUM;
DELETE;
END;

THEN DO;
PUT $0 "THE FOLLOWING INSTALLATION CANNOT BE ADDED"
"TO THE FILE, IT DOES NOT EXIST IN THE FILE:";
900 "STATE" 831 "STATE"
840 "CITY" 845 "CITY"
850 "COUNTY" 857 "COUNTY"
861 "INSTALLATION NUMBER" 861 INST_NUM;
DELETE;
END;