System Operations Studies for Automated Guideway Transit Systems

Detailed Station Model Programmer's Manual (Appendix)

John F. Duke
Roger Blanchard

GM Transportation Systems Division
General Motors Corporation
GM Technical Center
Warren MI 48090

January 1982
Final Report

US Department of Transportation
Urban Mass Transportation Administration

Office of Technology Development and Deployment
Office of New Systems and Automation
Washington DC 20590

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
UMTA-MA-06-0048-81-9		

4. Title and Subtitle	5. Report Date
SYSTEM OPERATIONS STUDIES FOR AUTOMATED GUIDEWAY TRANSIT SYSTEMS - Detailed Station Model Programmer's Manual, Appendix	January 1982
	6. Performing Organization Code: DTS-723
7. Author(s) John F. Duke, GM TSD, and Roger Blanchard, IBM Federal Systems Division	8. Performing Organization Report No.: DOT-TSC-UMTA-81-44,II
9. Performing Organization Name and Address: GM Transportation Systems Division* General Motors Corporation GM Technical Center Warren MI 48090	10. Work Unit No. (TRAIS): UM268/R2670
	11. Contract or Grant No.: DOT-TSC-1220-4
12. Sponsoring Agency Name and Address: U.S. Department of Transportation Urban Mass Transportation Administration Office of Technology Development and Deployment Office of New Systems and Automation Washington DC 20590	13. Type of Report and Period Covered: Final Report June 1977 - January 1979
	14. Sponsoring Agency Code: UTD-40

15. Supplementary Notes

*Under contract to: U.S. Department of Transportation
Research and Special Programs Administration
Transportation Systems Center
Cambridge MA 02142

16. Abstract

This appendix describes program design language for the maintenance and modification of this model, which is described in Report No. UMTA-MA-06-0048-81-8.

17. Key Words	18. Distribution Statement
Scheduled Service, Queue Size, Queue Time, Process Time, Demand-Responsive Single Party, Demand-Responsive Multiparty	DOCUMENT IS AVAILABLE TO THE PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	232	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

APPENDIX A

PROCESS DESIGN LANGUAGE

The program control and data flow for the IP, MP, and OP is most vividly depicted in Section 2, Hierarchical Figures, Section 5, Subprogram Logic Tables, and Appendix B, HIPO diagrams. The following PDL segments, listed alphabetically by module name, use natural language to define the operations on data from a logical viewpoint. The so-called outer syntax of PDL is defined by a few PDL keywords such as PROC, ENDPROC, IF, ELSE, ENDIF, CASE, ENDCASE, DOWHILE, DOUNTIL, and ENDDO, which have the same meaning as the equivalent PARAFOR statements. A tabular typographical format is used to segment and set-off related operations.

Then PDL segments are the most explicit program documentation other than the code itself. PDL should be referenced with the PARAFOR source code to gain a detailed understanding of program operation.

```
      PROJECT: AGT                    MEMBER: DAYTIM                    DATE:
      LIBRARY: DSM                    LEVEL:  01.00                     TIME:
      TYPE:    PLL                    USERID: XA3WED                    PAGE:
START
 COL  ---+-----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC: DAYTIM    <PROCEDURE TO GET DATE AND TIME>
    2    RUN TIME(DAYS,SECS)   <MACRO TO ACCESS SYSTEM CLOCK>
    3    FORMAT DATE TO:=<MM/DD/YY>
    3    FORMAT TIME TO:=<HHMM>
    3    RETURN
    1  ENDPROC
```

```
    PROJECT: AGT                    MEMBER: SAASYN
    LIBRARY: DSM                    LEVEL:  01.12
    TYPE:    PDL                    USERID: C120496
PART
COL ----+----1----+----2----+----3----+----4----+----5----+----6--
    PROC:SAASYN   <PROCESS ASYNCHRONOUS COMMANDS>                          000100
                                                                           000200
      WRITE NAME & TIME OF ASYNCH CARD TO BE EXEC NOW                      000300
    3 TEST FOR UNRECOGNIZED CARD                                           000400
    3 CASEENTRY(XMEVNT(X*))          <THE TYPE OF ASYNCHRONOUS EVENT       000500
   33                                OF THE XTN THAT HAS JUST COME         000600
   33                                OFF THE PEL>                          000700
                                                                           000800
    3   CASE(DATA/PARAM/OPTION/SELECT)                                     000900
    5     RUN GDIP TO REVISE SCI TYPE COMMONS                              001000
                                                                           001100
    3   CASE(FAILURE/RECOVERY/DEGRADATION/DEGRADATION RECOVERY - SL/TL)    001200
    5     RUN GDIP TO REVISE DATA ASSOCIATED WITH FAILURE                  001300
    5     RUN SAFAIL                                                       001400
                                                                           001500
    3   CASE(FLAGS)                                                        001600
    5     RUN SAFLAG                     <RESET DEBUG FLAGS>             001700
                                                                           001800
    3   CASE(TEXT)                                                         001900
    5     READ & WRITE ONE TEXT CARD                                       002000
                                                                           002100
    3   CASE(DEMAND CKPT)                                                  002200
    5     RUN CKPT                                                         002300
                                                                           002400
    3   CASE(END OF DECK/STOP)                                             002500
    5     CONVERT CURRENT ASYNCH XTN INTO STOP XTN                         002600
    5     SCHEDULE THIS STOP XTN                                           002700
                                                                           002800
    3   CASE(SIMULTANEOUS TRIP ARRIVALS)                                   002900
    5     FOR                                                              003000
    7       EACH TRIP ARRIVING SIMULTANEOUSLY                              003100
    5     DO                                                               003200
    7       GET TRIP XTN                                                   003300
    7       READ IN DATA(DESTINATION, SIZE)                                003400
            SET ORIGIN TO STATION BEING SIMULATED                          003500
            SET ARRIVAL TIME TO CLOCK                                      003600
            RUN SAVORG                                                     003700
    5     ENDDO                                                            003800
                                                                           003900
    3   CASE(SIMULTANEOUS VEH ARRIVALS)                                    004000
    5     FOR                                                              004100
    7       EACH VEH ARRIVING SIMULTANEOUSLY                               004200
    5     DO                                                               004300
    7       GET VEH XTN                                                    004400
    7       READ IN DATA(NEXT STOP, DIVERT TO STORE, SINK, ROUTE, NPASS,TRLEN)004500
    7       RUN SAVORG                                                     004600
    5     ENDDO                                                            004700
                                                                           004800
    3   CASE(INDEX)                                                        004900
    5     WRITE INDEX TEST CARDS TO INDEX FILE                             005000
                                                                           005100
    3   CASEELSE                                                           005200
    5     RUN ERROR("NOT RECOGNIZED")                                      005300
                                                                           005400
    3   ENDCASE                                                            005500
    3   READ IN NEXT COMMAND CARD                                          005600
    3   SET FLAG INDICATING THIS FILE USED (FOR INDEX PROCESSING)          005700
    3   SCHEDULE NEXT ASYNCH EVENT <FROM COMMAND CARD>                     005800
    1 ENDPROC                                                              005900
```

```
PROJECT: AGT                    MEMBER: SACKR
LIBRARY: DSM                    LEVEL:  01.02
TYPE:    PDL                    USERID: C120496

LABT
COL  ----+----1----+----2----+----3----+----4----+----5----+----6----

     PROC SACKR <CHECKPOINT AND RESTART PROCESSING>                          000100
       INITIALIZE ROUTINE WITH STARTING ADDRESS AND LENGTH OF ALL THE        000200
         COMMON AREAS THAT ARE TO BE WRITTEN AT CHECKPOINT OR RESTORED       000300
       AT RESTART                                                          000400
   1 SACKPT <PERFORM SYSTEM CHECKPOINTING>                                 000500
   6       <DSM CHECKPOINTING IS PERFORMED TO SAVE THE STATUS OF A SIM-    000600
   7        ULATION EXPERIMENT AT ANY POINT DURING THE SIMULATION RUN.     000700
   7        CHECKPOINTING CAN OCCUR AT PERIODIC INTERVALS OR VIA AN ASYN-  000800
   7        CHRONOUS DATA REQUEST. THE CHECKPOINT DATA CAN BE USED TO RE-  000900
   7        START THE SIMULATION BY REINITIALIZTION OF SYSTEM STATUS AS    001000
   7        SAVED BY THE CHECKPOINT. IN ORDER TO ALLOW THIS CAPABILITY     001100
   7        CHECKPOINT SAVES THE STATUS OF ALL DATA RELATED TO TRANSACTION 001200
   7        MANAGEMENT, STATION LINK, VEHICLE, TRIPS & SAMPLING. THE       001300
   7        ADDRESS OF THIS DATA IS RECORDED FOR CHECKPOINTING DURING      001400
   7        SYSTEM INITIALIZATION. THE ADDRESS OF THE CHECKPOINT/RESTART AREA 001500
   7        OBTAINED DURING SYSTEM INITALIZATION BY PROGAM SACKR.>         001600
   3   WRITE CHECKPOINT RECORD TO FILE                                     001700
   1 SAREST <PERFORM SYSTEM RESTART>                                       001800
   6       <SYSTEM RESTART IS PERFORMED WHEN A RESTART COMMAND IS RECOGNIZED 001900
   7        DURING SYSTEM INITIALIZATION. THE RESTART COMMAND MUST BE THE  002000
   7        FIRST ASYSNCHRONOUS DATA INPUT TO A SIMULATION RUN. SAREST     002100
   7        REINITIALIZES THE SIMULATION FOR RESUMPTION OF A PREVIOUS SIM- 002200
   7        ULATION EXPERIMENT IN WHICH A CHECKPOINT OF SYSTEM STATUS WAS  002300
   7        PERFORMED.>                                                    002400
   3   DO <READ CHECKPOINT FILE TO DESIRED TIME>                           002500
   5     READ CHECKPOINT RECORD                                            002600
   3   UNTIL                                                               002700
   5     TIME=RESTART TIME REQUESTED                                       002800
   3   ENDDO                                                               002900
   3   DO <REPOSITION TRIP FILE>                                           003000
   5     READ TRIP FILE                                                    003100
   3   UNTIL <FILE REPOSITIONED PROPERLY>                                  003200
   5     TIME OF RCD GT TIME OF CKPT & ORIGIN OF TRIP = STSIM              003300
   3   ENDDO                                                               003400
       FOR                                                                 003500
         THREE POTENTIAL VEHICLE ARRIVAL SOURCES(I)                        003600
   3   DO <REPOSITION VEHICLE FILE>                                        003700
   5     READ VEHICLE FILE(I)                                              003800
   3   UNTIL <FILE REPOSITIONED PROPERLY>                                  003900
   5     TIME OF RCD GT TIME OF CKPT                                       004000
   3   ENDDO                                                               004100
   3   DO <REPOSITION RUNTIME FILE>                                        004200
   5     READ RUNTIME FILE HEADER RECORD & ASSOCIATED FOLLOWERS            004300
   3   UNTIL <FILE REPOSITIONED PROPERLY>                                  004400
   5     TIME OF RCD GT TIME OF CKPT                                       004500
   3   ENDDO                                                               004600
   3   SET FLAG TO INDICATE WHICH MEMBER USED                              004700
   3   SCRATCH OLD TRIP, VEH & ASYNCH XTNS & SCHEDULE NEW ONES             004800
   1 ENDPROC                                                               004900
```

```
      PROJECT: AGT                    MEMBER: SACOMN                    DATE:
      LIBRARY: OSM                    LEVEL:  01.03                     TIME:
      TYPE:    PLL                    USERID: CI20496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--

    1  PROC:SACOMN
    6       USED TO FORCE AN ORDERING AT LINK EDIT TIME OF THE MP
    5       THAT IS IDENTICAL TO THAT OF THE IP
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SADADD                    DATE:
      LIBRARY: JCM                    LEVEL:  01.00                     TIME:
      TYPE:    PLL                    USERID: CI20496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC SADADD <INITIALIZE INPUT AREA ADDRESSES AND MESSAGE COMMONS>
    3    INITIALIZE ROUTINE WITH STARTING ADDRESSES OF INPUT COMMONS & LENGTH
    1  SANDTA
    3    READ BINARY SYSTEM DATA INTO INPUT COMMONS
    3    SET VALUES IN SCAMSG COMMON
    1  ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SAFAIL                    DATE:
      LIBRARY: GSM                    LEVEL:  01.00                     TIME:
      TYPE:    PCL                    USERID: C120496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SAFAIL
    2    <NOTE: SOME OF THE DATA USED HERE IS READ IMMEDIATELY
   10            BEFORE SAFAIL IS CALLED>
    3    IF
    5      SL_RELATED
    3    THEN
    5      SL*=AFAIL(2)
    5      CASEENTRY(TYPE OF ACTIVITY)

    5      CASE(SL FAILURE)
    7        CASEENTRY(FAILURE LOCATION)
    7        CASE(ENTRY)
    9          SLENT(SL*):=1              <MARK ENTRY OF SL* AS BLOCKED>
    7        CASE(EXIT)
    9          SLEXIT(SL*):=1             <MARK EXIT OF SL* AS BLOCKED>
    7        ENDCASE

    5      CASE(SL FAILURE RECOVERY)
    7        CASEENTRY(RECOVERY LOCATION)
    7        CASE(ENTRY)
    9          SLENT(SL*):=0              <MARK ENTRY OF SL* AS ACTIVE>
    9          RUN SSPMAC(SL*,UPSTREAM) <TRY TO GET VEHICLES MOVING UPSTREAM
   37                                        OF SL*>
    7        CASE(EXIT)
    9          SLEXIT(SL*):=0             <MARK EXIT OF SL* AS ACTIVE>
    9          RUN SUPMAC(SL*,SELF)     <TRY TO GET VEHICLES MOVING ON SL*>
    7        ENDCASE

    5      CASE(SL DEGRADE/DEGRADE RECOVERY)
    7        SLPENT(SL*):=NEW PENALTY FACTOR <REVISE TRAV DELAY FACT ON SL*>

    5      ENDCASE

    3    ELSE      <TL RELATED>
    5      TL*=AFAIL(2)
    5      <TL FAILURE/FAILURE RECOVERY/DEGRADE/DEGRADE RECOVERY>
    7        IF
    9          FAILURE RECOVERY
    7        THEN
    8          RUN SUPMAC(TL*)          <TRY TO GET TRIPS MOVING UPSTREAM
   37                                        OF TL*>
    7        ENDIF
    3    ENDIF
    1  ENDPROC
```

```
      PROJECT: AUT                    MEMBER: SAFINM                  DATE:
      LIBRARY: SRM                    LEVEL:  01.01                   TIME:
      TYPE:    PDL                    USERID: C120496                 PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC: SAFINM <FINAL MODEL REPORT>
    3    PRINT RAW STATISTICS
    5    OPTIONALLY PRINT INTERNAL VARIABLES DESCRIBING SL'S & TL'S
    1  ENDPROC
```

```
    PROJECT: AGT                    MEMBER: SAFINS
    LIBRARY: DSM                    LEVEL:  01.02
    TYPE:    PDL                    USERID: C120496
PART
LOL ----+----1----+----2----+----3----+----4----+----5----+----6--

    PROC SAFINS <FINAL SYSTEM REPORT>                                          000100
      WRITE REPORT HEADER                                                      000200
  3   FOR                                                                      000300
  5     EACH OF THE 10 HISTOGRAM INTERVALS                                     000400
  3   DO                                                                       000500
  5     WRITE INTERVAL NUMBER,UPPER EDGE OF HISTOGRAM INTERVAL, HISTOGRAM      000600
  7       VALUE & PERCENTAGE OF TOTAL SCHEDULED EVENTS                         000700
  3   ENDDO                                                                    000800
  3   RUN SAWTIW(EP-SAWTIX) TO LIST MEMBERS USED IN INDEX FILE                 000900
  1 ENDPROC                                                                    001000
```

```
      PROJECT: AUT                    MEMBER: SAFLAG                    DATE:
      LIBRARY: USM                    LEVEL:  01.01                     TIME:
      TYPE:    PLI                    USERID: C120496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SAFLAG <RESET ALL DEBUG FLAGS>
    2    SET ALL FLAG VALUES FALSE
    3    UNTIL
    4      A ZERO FLAG FIELD FOUND ON A CARD
    3    DO
    5      SET STRINGS OF CONSECUTIVE FLAGS TRUE
    3    ENDDO
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SAINIT                    DATE:
      LIBRARY: DSM                    LEVEL:  01.10                     TIME:
      TYPE:    PDL                    USERID: CI20496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC SAINIT <PERFORM SYSTEM INITIALIZTION>
    2    ESTABLISH INTERRUPT HANDLER INTERFACE
    3    OBTAIN SYSTEM STATUS AREA ADDRESSES <SANTSA>
    4      <ADDRESS OF AREA TO BE INITIALIZED FROM INPUT PROCESSOR DATA
    5       CONTAINED IN AGT.STRUC.SYSTEM & ADDRESS OF THE AREA CONTAINING
    6       ALL MODEL PROCESSOR DATA WHICH MUST BE CHECKPOINTED OR RE-
    7       INITIALIZED IN THE CASE OF A RESTART REQUEST>
    8    READ ASYNCHRONOUS DATA INPUT HEADER
    9    IF <A SYSTEM RESTART REQUIRED>
   10      HEADER.REST
   11    THEN <RESTART SIMULATION FROM A CHECKPOINT>
   12      RUN SARES1 <PERFORM SYSTEM RESTART>
   13      READ NEXT ASYNCHRONOUS DATA INPUT HEADER
   14    ELSE <PERFORM INITIALIZATION OF SIMULATION EXPERIMENT>
   15      INITIALIZE SYSTEM CHARACTERISTICS & ERROR ROUTINE INTERFACE<SANDTA>

   16      INITIALIZE TRANSACTION DATA <SANXIN>

   17      INITIALIZE FUTURE EVENTS LIST <SANFEL>

   18      INITIALIZE STATION & TRIP LINK MODELS <SANMDL>

   19      DO <INITIALIZE SYSTEM LEVEL EVENTS>

   20        GET A SYSTEM TRANSACTION <XACTIVE>
   21        READ FIRST TRIP RECORD
   22        XSEVNT(XACTIVE):=TRIP ORIGINATION EVENT
   23        RUN SAFEL <SCHEDULE TRIP ORIGINATION TRANSACTION>

   24        IF
   25          PERIODIC SAMPLING
   26        THEN
   27          GET A SYSTEM TRANSACTION
   28          RUN SAFEL <SCHEDULE FIRST PERIODIC SAMPLE EVENT>
   29        ENDIF

   30        IF
   31          PERIODIC CHECKPOINTING REQUIRED
   32        THEN
   33          GET A SYSTEM TRANSACTION
   34          RUN SAFEL <SCHEDULE FIRST PERIODIC SAMPLING EVENT>
   35        ENDIF

   36        FOR
   37          THREE POTENTIAL VEHICLE ARRIVAL SOURCES
   38        DO <SCHEDULE FIRST TRIP ARRIVAL>
   39          IF
   40            SOURCE IS REQUIRED
   41          THEN
   42            GET A SYSTEM TRANSACTION <XACTIVE>
```

```
      PROJECT: AGT                    MEMBER: SAINIT                    DATE:
      LIBRARY: CSM                    LEVEL:  01.10                     TIME:
      TYPE:    PGL                    USERID: C12049                    PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  11            READ FIRST VEHICLE RECORD
  11            XSEVNT(XACTIVE):=VEHICLE ORIGINATION EVENT
  11            RUN SAPPEL <SCHEDULE VEHICLE ORIGINATION TRANSACTION>
   9          ENDIF
   7        ENDDO

   5      ENDDO

   5      UPDATE INDEX FILE <SAUPIX>

   5      <INITIALIZE FIRST ASYNCHRONOUS SIM INPUT>
   5      GET A SYSTEM TRANSACTION <XACTIVE>
   5      XSEVNT(XACTIVE):=ASYNCHRONOUS DATA EVENT
   5      RUN SAPPEL <SCHEDULE ASYNCHRONOUS EVENT>

   3    ENDIF
   1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SAMAIN                    DATE:
      LIBRARY: DSM                    LEVEL:  01.27                     TIME:
      TYPE:    SIL                    USERID: C120496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+---7-

    1  PROC:SAMAIN   <ARCHITECTURE MAIN PROCEDURE FOR DSM>
    5      <SAMAIN IS PASSED CONTROL TO BEGIN SIMULATOR EXECUTION AFTER THE
    6       SAMPLING & CHECKPOINT FILE NAMES PASSED VIA THE PARM FIELD
    6       IN THE EXECUTION PROCEDURE HAS BEEN SAVED BY PROGRAM SANTIA
    6       FOR LATER USE IN UPDATING THE INDEX FILE DURING SYSTEM INITIAL-
    6       IZATION>
    3     RUN SAINIT                          <INITIALIZE SIMULATOR>
    3     WHILE
    5       #LOOPS WITH CLOCK UNCHANGED<CLOOP
    5     DO
    5       RUN SAFEL    (GET NEXT EVENT OFF FEL (X*))
    5       UPDATE CLOCK TO TIME OF THIS EVENT

    5       CASEENTRY(XSEVNT(X*))           <SYSTEM EVENT TYPE>

    5       CASE(VEHICLE EVENT)
    7         RUN SASCTL                    <STATION LINK CONTROL>

    5       CASE(TRIP EVENT)
    7         RUN SAUCTL                    <TRIP LINK CONTROL>

    5       CASE(ASYNCHRONOUS EVENT)
    7         RUN SAASYN                    <READ ASYNCHRONOUS DATA>

    5       CASE(SAMPLING FOR OUTPUT)
    7         RUN SASAMP                    <OUTPUT SAMPLE>
    7         SCHEDULE NEXT SAMPLING

    5       CASE(PERIODIC CHECKPOINT)
    7         RUN SACKPT                    <TAKE A CHECKPOINT OF THE SYSTEM>
    7         SCHEDULE NEXT PERIODIC CHECKPOINT

    5       CASE(TRIP ORIGINATION)
    7         RUN SATORG & SATRD            <TRIP ARRIVES AT STATION>
    7         SCHEDULE NEXT TRIP ORIGINATION

    5       CASE(VEHICLE ORIGINATION)
    7         RUN SAVORG & SAVRD            <VEHICLE ARRIVES AT STATION>
    7         SCHEDULE NEXT VEHICLE ORIGINATION

    5       CASE(SL PROMPT)
    7         RUN SASPRM(SL*,FLAG)        <GET QUEUED VEHICLES MOVING>
    7         FREE XTN(X*) THAT JUST CAME OFF THE FEL

    5       CASE(TL PROMPT)
    7         RUN SAUPRM(TL*)           <GET QUEUED TRIPS MOVING>
    7         FREE XTN(X*) THAT JUST CAME OFF THE FEL

    5       CASE(END OF SIM EVENT)
    7         RUN SAFINM
```

```
      PROJECT: AGI                    MEMBER: SAMAIN                  DATE:
      LIBRARY: DSM                    LEVEL:  01.27                   TIME:
      TYPE:    PLI                    USERID: C120496                 PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    7          RUN SAFINS

    5        INCLASS

    3     ENDJOB
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SANFEL                    DATE:
      LIBRARY: DSM                    LEVEL:  01.00                     TIME:
      TYPE:    PCL                    USERID: C120496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC SANFEL <INITIALIZE FUTURE EVENT LIST>
   3    GET A SYSTEM SERVICE XTN
   3    INITIALIZE IT TO BE INFINITE TIME MULTIPLE THREAD XTN
   3    ZERO OUT CLOCK TABLE
   3    INITIALIZE SCMFEL VARIABLES
   1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SANMDL                    DATE:
      LIBRARY: SSM                    LEVEL:  00.06                     TIME:
      TYPE:    PCL                    USERID: CI20496                   PAGE:
START
 COL  ----#----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC SANMDL <INITALIZE SIMULATION MODEL>
    2    FOR <ALL STATION LINKS>
    3      <NSL
    4    DO
    5      SET ALL SCMSL VARIABLES
    6    ENDDO
    7    FOR <ALL TRIP LINKS>
    8      <NTL
    9    DO
   10      SET ALL SCMTL VARIABLES
   11   ENDDO
   12   FOR <ALL VEHICLES>
   13     <NV
   14   DO
   15     SET ALL SCMT VARIABLES
   16   ENDDO
   17   FOR <ALL TRIPS>
   18     KNT
   19   DO
   20     SET ALL SCMV VARIABLES
   21   ENDDO
   22   SET SCMSYS VARIABLES
   23   RUN SAZNIT
   24 ENDPROC
```

```
      PROJECT: AGI                    MEMBER: SANSAV                    DATE:
      LIBRARY: SRM                    LEVEL:  01.01                     TIME:
      TYPE:    PLL                    USERID: C120496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SANSAV <INITIALIZE CHECKPOINTING AND SYSTEM DATA READ PROCESSES>
    3    CONVERT THE ADDRESS OF THE ADDRESS OF THE START OF
    5      ALL THE COMMONS TO DIRECT ADDRESS OF THE START
    3    SET REGISTER 1 TO POINT TO THIS STARTING ADDRESS AND THE LENGTH
    5      OF ALL THE COMMONS
    7    RUN SACKR
    3    CONVERT THE ADDRESS OF THE ADDRESS OF THE START OF THE INPUT
    5      COMMONS TO DIRECT ADDRESS OF THE START
    3    SET REGISTER 1 TO POINT TO THIS STARTING ADDRESS AND THE LENGTH
    5      OF THE INPUT COMMONS
    7    RUN SADATD
    1  ENDPROC
```

```
     PROJECT: AGT                    MEMBER: SANTIX                  DATE:
     LIBRARY: USM                    LEVEL:  01.00                   TIME:
     TYPE:    PUL                    USERID: C120496                 PAGE:
START
  COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC SANTIX <INITIALIZE MEMBER NAME STRING>
    3    SAVE ADDRESS OF CONTENTS OF PARM FIELD
    3    RUN SAMAIN
    1  SAUPTR <PASS MEMBER NAME STRING TO SAWTIX>
    3    RESTORE ADDRESS OF PARM FIELD TO REGISTER 1
    3    RUN SAWTIX
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SANTSA                    DATE:
      LIBRARY: DSM                    LEVEL:  01.00                     TIME:
      TYPE:    PDL                    USERID: C120496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC SANTSA
    3    CONVERT THE ADDRESS OF THE ADDRESS OF THE MODEL COMMONS, THE
    3      ADDRESS OF THE ADDRESS OF THE INPUT COMMONS, AND THE ADDRESS
    5      OF THE ADDRESS OF THE END OF THE COMMONS INTO LENGTHS
    3    RUN SANSAV WITH ADDRESSES AND LENGTHS
    1  ENDPROC
```

```
      PROJECT: AOI                    MEMBER: SANXIN                  DATE:
      LIBRARY: USM                    LEVEL:  01.00                   TIME:
      TYPE:    PLI                    USERID: C120496                 PAGE:
STMT
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC (SANXIN) <INITIALIZE TRANSACTION DATA>
    2    /* CLEAR OUT VARIABLES OF SCMXTN
    3    SET SEARCH UP EACH TO POINT TO NEXT
    4    ESTABLISH HEADS OF VEHICLE,TRIP & SYSTEM SERVICE XTN AVAILABLE LISTS
    5  END PROC
```

```
      PROJECT: ACT                    MEMBER: SAPFEL                    DATE:
      LIBRARY: SEM                    LEVEL:  00.16                     TIME:
      TYPE:    PLL                    USERID: P326507                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC SAPFEL <PLACE TRANSACTION ON FUTURE EVENTS LIST>
    6       <SAPFEL PERFORMS THE SCHEDULING OF TRANSACTION ON THE FUTURE EVENT
    7        LIST. SAPFEL IS INVOKED BY EITHER THE SCHEDULING OF A TRANACTION
    7        VIA THE SCHED MACRO OR VIA A DIRECT CALL. THE TRANSACTION TO BE
    7        PLACED ON THE FUTURE EVENTS LIST IS EITHER PLACED IN THE CLOCK
    7        TABLE OR ON THE MULTIPLE THREAD LIST DEPENDING UPON WHETHER THE
    7        SCHEDULE TIME IS WITHIN THE CURRENT CLOCK TABLE INTERVAL OR AT
    7        SOME EXTENDED TIME IN THE FUTURE. SCHEDULING ON THE CLOCK TABLE
    7        INVOLVES FINDING THE CORRECT POSITION FOR INSERTION & ADDING THE
    7        TRANSACTION ID TO THE CLOCK TABLE. MULTIPLE THREAD SCHEDULING
    7        REQUIRES EITHER THE ADDITION OF THE TRANSACTION TO AN EXISTING
    7        MULTIPLE THREAD LOOP OR THE CREATION OF A NEW MULTIPLE THREAD
    7        LOOP. CONCURRENT WITH SCHEDULING TRIP AND VEHICLE TRANSACTIONS,
    7        TRIP NEXT EVENT DATA IS WRITTEN TO THE TRIP AND VEHICLE FILE WHEN
    7        REQUIRED.  A HISTORY OF THE TRIP'S/VEHICLE'S LAST QUELED STATUS I
    7        ALSO WRITTEN TO THE FILE.>
    3    IF (DELTA.LT.0)
    5      SCHEDULING TIME < 0
    3    THEN
    5      RUN ERROR <'ATTEMPT TO SCHEDULE AN EVENT FOR A NEGATIVE TIME
   17                 DELAY = ZERO ASSUMED'>
    3    ENDIF
    3    <COLLECT FEL STATS AND WRITE TO TRIP AND VEHICLE FILE>
    3    RUN SAPFEL2
    6       <DETERMINE IF SCHEDULING REQUIRED IN CLOCK TABLE OR MULTIPLE
    7        THREAD REQUIRED>
    3    CPOS=RELATIVE TIME POSITION IN CLOCK TABLE INTERVALS
    3    IF <NUMBER OF INTERVALS WITHIN CLOCK TABLE SPAN>
    5      CPOS.LT.CLSIZE
    3    THEN <SCHEDULE TRANSACTION WITHIN CLOCK TABLE>
    5      CPOS=CPOS+CLPOS
    5      IF <POSITION WRAPS AROUND TABLE>
    7        CPOS.GT.CLSIZE
    5      THEN <SET INDICATOR FOR RESCAN >
    7        CLSCAN=1
    5      ENDIF
    5      <INSERT TRANSACTION IN CLOCK TABLE IN PROPER POSITION>
    5      IF <CLOCK TABLE INTERVAL EMPTY>
    7        CLTABL(CPOS)=0
    7        MOVE TRANSACTION IN TABLE
    5      ELSE
    7        MOVE TRANSACTION IN TIME ORDERED POSITION OF CLOCK INTERVAL
    5      ENDIF
    3    ELSE <MULTIPLE THREAD SCHEDULING REQUIRED. DETERMINE IF MULTIPLE
    8         THREAD LOOP MUST BE CREATED OR ONE EXISTS>
    5      FOR
    7        EACH MULTIPLE THREAD TRANSACTION
    5      DO <DETERMINE IF TRANSACTION BELONG IN LOOP>
    7        IF
    9          TIME OF MULTIPLE*THREAD<SCHEDULE TIME & SCHEDULE TIME      WITHIN
```

```
      PROJECT: AGT                        MEMBER: SAPFEL                        DATE:
      LIBRARY: SRM                        LEVEL:  00.16                         TIME:
      TYPE:    PEL                        USERID: P326807                       PAGE:
START
 COL  ---+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    4          MULTI-LC THREAD LOOP SIZE
    7        THEN <PLACE TRANSACTION IN THIS MULTIPLE THREAD LOOP>
    8          MOVE TRANSACTION IN MULTIPLE THREAD LOOP
    7        ELSE <A NEW MULTIPLE THREAD LOOP MUST BE CREATED>
    8          GET A SYSTEM TRANSACTION
    4          MOVE TRANSACTION IN MULTIPLE THREAD LOOP
    9          MOVE TRANSACTION TO BE SCHEDULED IN MULTIPLE THREAD CHAIN
    7        ENDIF
    5      ENDDO
    3    ENDIF
    3    RUN SAPFEL3 <IF VEHICLE GOING ON FEL IS LEAD VEHICLE OF A TRAIN,
   10                _      UPDATE VTIME OF FOLLOWER VEHICLES TO VTIME OF LEAD
   15                       VEHICLE>
    1  PROCEND
```

```
      PROJECT: AUT                    MEMBER: SAPFEL2                    DATE
      LIBRARY: GSM                    LEVEL:  01.04                      TIME
      TYPE:    PEL                    USERID: P32507                     PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC: SAPFEL2  <WRITE TRIP AND VEHICLE OUTPUT FILE AND UPDATE QUEUED
   17                   AND FEL STATISTICS>
    1  ------------------------------------------------------------------------
    1  LOCAL VAR NAME  DIM    DESCRIPTION
    1  --------------- ------ ---------------------------------------------
    1  WASQD           -      INDICATES A VEHICLE OR TRIP HAD BEEN QUEUED
   26                           PRIOR TO GOING ON THE FEL THIS TIME
   26                           T===>TRANSACTION WAS QUEUED
   26                           F===>TRANSACTION CAME OFF THE FEL
    1  ------------------------------------------------------------------------
    3    WASQD:=F
    3    IF
    5      THE TRANSACTION IS A VEHICLE
    3    THEN
    5      IF
    7        THE TRANSACTION IS NOT A NEW VEHICLE
    7        VTIME (V) =: (-1)
    5      THEN
    7        IF
    9          CLOCK > VTIME(V)/10
    7        THEN
    9          UPDATE STATS FOR VEHICLE LEAVING THE QUEUED STATE
    9          WASQD:=T
    7        ELSE
    9          IF
   11            V = VACTIV
    9          THEN
   11            UPDATE STATS FOR A VEHICLE LEAVING THE FEL STATE
    9          ELSE
   11            IF
   13              VSEVCH (V) = KMX + 1
   13              THIS VEHICLE WAS FORMERLY ON THE TRAIN OF AN ACTIVE
   13              VEHICLE
   11            THEN
   13              UPDATE STATS FOR A VEHICLE LEAVING THE FEL STATE
   13              VSEVCH(V) = 0
   11            ELSE
   13              UPDATE STATS FOR A VEHICLE LEAVING THE QUEUED STATE
   13              WASQD:=T
   11            ENDIF
    9          ENDIF
    7        ENDIF
    5      ENDIF
    5      IF
    7        TRIP AND VEHICLE WRITTEN OUTPUT FILE OPTION IS IN EFFECT
    7        ATVF:=T
    5      THEN
    7        IF
    9          WASQD = T
    7        THEN
```

```
      PROJECT: AGI                    MEMBER: SAMPSL2                    DATE:
      LIBRARY: SSM                    LEVEL:  01.04                      TIME:
      TYPE:    PSL                    USERID: P326807                    PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    5           WRITE A QUEUED RECORD
    6           WRITE AN ACTIVITY RECORD
    7         ELSE
    8           WRITE AN ACTIVITY RECORD
    7         ENDIF
    5       ENDIF
    5       WRITE STATS FOR A VEHICLE ENTERING THE FEL STATE
    3     ELSE
    6       IF
    7         THE TRANSACTION IS A TRIP
    5       THEN
    7         IF -
    8           THE TRANSACTION IS NOT A NEW TRIP
    9           TTIME(T) ¬= (-1)
    7         THEN
    8           IF
   11             CLOCK = TTIME(T)/10 AND
   11             T = TACTIV
    9           THEN
   11             UPDATE STATS FOR THE TRIP LEAVING THE FEL STATE
    9           ELSE
   11             MASGD:= T
   11             UPDATE STATS FOR A TRIP LEAVING THE QUEUED STATE
    6           ENDIF
    7         ENDIF
    7         IF
    8           TRIP AND VEHICLE WRITTEN OUTPUT FILE OPTION IS IN EFFECT
    8           ATVF:=T
    7         THEN
    8           IF
   11             MASGD = T
    5           THEN
   11             WRITE A QUEUED RECORD
   11             WRITE AN ACTIVITY RECORD
    9           ELSE
   11             WRITE AN ACTIVITY RECORD
    4           ENDIF
    7         ENDIF
    7         UPDATE STATS FOR A TRIP ENTERING THE FEL
    5       ENDIF
    3     ENDIF
    1   ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SAPFEL3                    DATE:
      LIBRARY: DSM                    LEVEL:  01.00                      TIME:
      TYPE:    PCL                    USERID: P326507                    PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SAPFEL3 <IF THE LEAD VEHICLE OF THE TRAIN IS PUT ON THE FEL,
   15                UPDATE VTIME FOR THE FOLLOWING VEHICLES>
    3    IF
    5      THE TRANSACTION IS A VEHICLE AND
    6      THE VEHICLE IS THE LEAD VEHICLE OF A TRAIN
    3    THEN
    5      FOR
    7        EACH VEHICLE IN THE TRAIN
    5      DO
    7        VTIME(VID):= VTIME(V)
    5      UNTIL
    7        ALL VEHICLES IN THE TRAIN HAVE BEEN UPDATED
    5      ENDDO
    3    ENDIF
    1  ENDPROC
```

```
PROJECT: AGT                    MEMBER: SAMFEL                    DATE:
LIBRARY: SSM                    LEVEL:  00.03                     TIME:
TYPE:    PEL                    USERID: M039547                   PAGE:
START
COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  1  PROC SAMFEL <OBTAIN A TRANSACTION FROM THE FUTURE EVENTS LIST>
  2       <SAMFEL IMPLEMENTS THE CLOCK TABLE & MULTIPLE THREAD FEL AL-
  3        GORITHMS. THE NEXT TRANSACTION REQUIRING THE NEXT EVENT TO BE
  4        PERFORMED (NEXT MUST IMPLEMENT) IS OBTAINED FROM EITHER THE CUR-
  5        RENT CLOCK TABLE INTERVAL TRANSACTION LIST OR FROM THE NEXT
  6        NON-EMPTY CLOCK TABLE INTERVAL FOUND BY PERFORMING A SEQUENTIAL
  7        SCAN OF THE CLOCK TABLE. IF A TRANSACTION IS NOT FOUND DURING THE
  8        SCAN (END OF CLOCK TABLE REACHED), THE NEXT MULTIPLE THREAD FEL
  9        LIST IS USED TO RELOAD THE CLOCK TABLE & FIRST RELOADED TRANS-
 10        ACTION IS SELECTED.>
 11    DO <GET NEXT TRANSACTION>
 12      XLAST:=CLTABL(CLPOS)
 13      IF <NO TRANSACTIONS SCHEDULED IN THIS INTERVAL>
 14        XLAST:=0
 15      THEN <FIND INTERVAL IN WHICH TRANSACTION PROCESSING IS REQUIRED>
 16        XPOS:=CLPOS
 17        DO
 18          CLMINI:=CLMINI+CLSMAL
 19          XLAST:=CLTABL(CLPOS)
 20        UNTIL <NONE EMPTY INTERVAL FOUND OR END OF CLOCK TABLE REACHED>
 21          XLAST>0 OR CLPOS=CLSIZE
 22        ENDDO
 23      ENDIF
 24      IF <TRANSACTION NOT FOUND IN CLOCK TABLE>
 25        XLAST=0
 26      THEN <RELOAD CLOCK TABLE FROM NEXT MULTIPLE THREAD TRANSACTION>
 27        MTHRD:=FIRST MULTIPLE THREAD TRANSACTION
 28        DO <UNLOAD NEXT M/T LOOP>
 29          IDMT:=MTHRD
 30          CLBASE:=XTIME(MTHRD)
 31          CLMINI:=CLBASE
 32          WHILE
 33            TRANSACTION REMAIN IN MULTIPLE THREAD LIST
 34          DO <PEACE TRANSACTION IN CLOCK TABLE>
 35            TIME:=XTIME(IDMT)
 36            POS:=1+(TIME-CLMINI)/CLSMAL
 37            IF <NO OTHER TRANSACTIONS SCHEDULED IN INTERVAL>
 38              CLTABL(CLPOS)=0
 39            THEN
 40              XCHAIN(IDMT)=IDMT
 41              CLTABL(POS):=IDMT
 42            ELSE<INSERT TRANSACTION IN TIME ORDER>
 43              LAST:=CLTABL(CLPOS)
 44              DO <FIND POSITION FOR XTN ID CHAINING>
 45                IF <TIME > THAN TIME FOR THIS TRANSACTION OR LAST IN
 46                    CLOCK TABLE INTERVAL>
 47                  XTIME(LAST)>XTIME(IDMT)OR XCHAIN(LAST)=LAST
 48                THEN <INSERT XTN>
 49                  XCHAIN(IDMT):=XCHAIN(LAST)
 50                  XCHAIN(LAST):=IDMT
```

```
      PROJECT: AGT                    MEMBER: SAMPEL                    DATE:
      LIBRARY: DSM                    LEVEL:  C0.03                     TIME:
      TYPE:    PCL                    USERID: H639547                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  17                  CLTABL(POS):+IDMT
  17                  LAST:=0
  15                ELSE<TRY NEXT XTN IN CURRENT CLOCK TABLE INTERVAL>
  17                  LAST:=XCHAIN(LAST)
  15                ENDIF
  13              UNTIL <XTN INSERTED IN CLOCK TABLE>
  15                LAST=0
  13              ENDDO
  11            ENDIF
   9          ENDDO
   9          FREE MULTIPLE THREAD TRANSACTION
   9          XPOS:=XPOS-CLSIZE <FIRST INTERVAL IN CLOCK TABLE>
   7        ENDDO
   5      ENDIF
   3    UNTIL <TRANSACTION FOUND OR END OF MULTIPLE THREAD>
   5      LAST>0
   3    ENDDO
   3    XACTIVE:=LAST
   1  PROCEND
```

```
    PROJECT: AGT                        MEMBER: SASAMP
    LIBRARY: DSM                        LEVEL:  01.04
    TYPE:    PDL                        USERID: C120496
PART
COL ----+----1----+----2----+----3----+----4----+----5----+----6-

    PROC SASAMP <PERFORM PERIODIC SAMPLING EVENT>                          000100
         <PERIODIC SAMPLING INVOLVES THE RECORDING OF SYSTEM STATUS &       000200
          MODELLING STATISTICS IN THE RAW STATISTICS FILE>                  000300
  3   IF                                                                   000400
  5     SNAPSHOT REPORT REQUIRED                                           000500
  3   THEN                                                                 000600
  5     RUN SAFINM                                                         000700
  3   ENDIF                                                                000800
                                                                           000900
  3   RUN SZINT    <END-POINT ADJUST INTEGRALS>                            001000
                                                                           001100
  3   GATHER STATISTICS FOR AVERAGE TIME NUMBERS FOR PERFORMANCE SUMMARY   001200
                                                                           001300
  3   WRITE STATION-WIDE STATISTICS HEADER                                 001400
  3   WRITE STATION-WIDE STATISTICS FOLLOWER RECORDS(1 PER STATION STATE)  001500
                                                                           001600
  3   WRITE STATION LINK STATISTICS HEADER                                 001700
  3   WRITE STATION LINK STATISTICS FOLLOWER RECORDS(1 PER SL STATE)       001800
                                                                           001900
  3   WRITE TRIP LINK STATISTICS HEADER                                    002000
  3   WRITE TRIP LINK STATISTICS FOLLOWER RECORDS(1 PER TL STATE)          002100
                                                                           002200
  3   RUN SZZERO       <RESET STATISTICS>                                  002300
  1 ENDPROC                                                                002400
```

```
      PROJECT: ACT                    MEMBER: SASCTL                    DATE
      LIBRARY: USM                    LEVEL:  01.07                     TIME
      TYPE:    PLL                    USERID: P326507                   PAGE
START
 COL  ---------1---------2---------3---------4---------5---------6---------7

    1  PROC: SASCTL(VACTIV) <STATION LINK CONTROL--RUN AT VEHICLE EVENT TIME>
    2    RUN SSMOD(VACTIV) <PUT V THROUGH THE MODEL OF THE SL>
    3    IF
    4      ADONESET  <DONE WITH ALL EVENTS ON THE SL OF V>
    5    THEN  <TRY TO MOVE V TO ITS NEXT SL>
    6      RUN SSTEST(VACTIV) <FIND NEXT SL AND TEST IF CAN ENTER>
    7      IF  <CAN_ENTER NEXT SL>
    8        AENTRSET
    9      THEN
   10        RUN SSLEAV(VACTIV) <LEAVE CURRENT SL>
   11        IF  <LEAVING MODELED AREA>
   12          SL=SINK
   13        THEN
   14          RUN SASCTL1(VACTIV)
   15          <ACCUMULATE STATS. ON TRIPS AND VEHICLES>
   16          ACCUMULATE STATS ON VEHICLE LEAVNG THE FEL ONLY
   17          <RETURN TRIPS AND VEHICLES TO AVAILABILITY CHAINS>
   18        ELSE  <NEXT SL IS NOT A SINK BUT RATHER ANOTHER STATION LINK>
   19          VSL(VACTIV):=AVNXSL <SET CURRENT SL TO NEXT SL>
   20          VMEVNT(V):=0
   21          <RESET THE VEHICLE EVENT OF VACTIVE TO NO EVENTS DONE YET>
   22          RUN SSMOD(VACTIV) <ENTER NEW SL FOR INITIAL PROCESSING>
   23        ENDIF
   24      ENDIF
   25    ELSE <VEHICLE STILL HAS EVENTS ON THE SL AND SSMOD PUT THE VEHICLE
   26           ON THE FEL>
   27    ENDIF
   28    IF
   29      VOREAD (VACTIV) ¬= 0  OR
   30      VMEVNT (VACTIV) = 6
   31    THEN
   32      ACCUMULATE STATS ON VEHICLE LEAVING THE FEL
   33      ACCUMULATE STATS ON VEHICLE ENTERING THE QUEUED STATE
   34    ENDIF
   35  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SASCTL1                    DATE:
      LIBRARY: GSM                    LEVEL:  01.04                      TIME:
      TYPE:    MSL                    USERID: P326507                    PAGE:
START
  COL  ---+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SASCTL1  <RETURN VEHICLE AND TRIP TRANSACTIONS TO THEIR AVAIL-
   10                 ABILITY CHAINS AND COLLECT STATISTICS>
    3    IF
    5      THE TRAIN IS A SINGLE VEHICLE
    3    THEN
    5      SET THE VTRNCH OF THE VEHICLE TO POINT TO ITSELF
    3    ENDIF
    3    <DEFINE A HEAD TO THE VEHICLE TRAIN CHAIN SO THAT DEQUEUING CAN
    4     BE DONE USING THE DQUE MACRO>
    3    QHEAD:=V
    3    PASSCNT:=0
    3    DO    -
    6      DQUE THE FIRST VEHICLE
    5      PASSCNT = PASSCNT + VNPASS (V)
    3      WHILE
    7        THERE ARE TRIPS ON THE VEHICLE
    6      DO
    7        DQUE THE FIRST TRIP
    7        FREE THE TRIP
    7        COLLECT STATISTICS FOR THE TRIP RIDING OUT OF THE STATION
    8      ENDDO
    5      FREE THE VEHICLE
    5      COLLECT STATISTICS FOR THE VEHICLE RIDING OUT OF THE STATION
    3    UNTIL
    5      THE TRAIN CHAIN IS EXHAUSTED
    3    ENDDO
    3    IF
    5      STATION LINK SINK = GUIDEWAY EXIT
    3    THEN
    5      COLLECT STATISTICS FOR VEHICLES LEAVING THE STATION FROM THE
    5      GUIDEWAY
    5      COLLECT STATISTICS OF AVERAGE NUMBER OF PASSENGERS ON VEHICLE
    5      LEAVING THE STATION FROM THE GUIDEWAY
    3    ELSE
    5      IF
    7        STATION LINK SINK = MODAL EXIT BEFORE PROCESSING
    5      THEN
    7        COLLECT STATISTICS FOR VEHICLES LEAVING THE STATION FROM THE
    7        MODAL EXIT BEFORE PROCESSING
    7        COLLECT STATISTICS OF AVERAGE NUMBER OF PASSENGERS ON VEHICLE
    7        LEAVING THE STATION FROM THE MODAL EXIT BEFORE PROCESSING
    5      ELSE
    7        COLLECT STATISTICS FOR VEHICLES LEAVING THE STATION FROM THE
    7        MODAL EXIT AFTER PROCESSING
    7        COLLECT STATISTICS OF AVERAGE NUMBER OF PASSENGERS ON VEHICLE
    7        LEAVING THE STATION FROM THE MODAL EXIT AFTER PROCESSING
    5      ENDIF
    3    ENDIF
    1  ENDPROC
```

```
    PROJECT: AGT                    MEMBER: SASPRM                    DATE:
    LIBRARY: DSM                    LEVEL:  01.21                     TIME:
    TYPE:    PL1                    USERID: CI20496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  1  PROC:SASPRM(SL#,FLAG)
  2    <THIS SEGMENT IS CALLED WHEN IT MIGHT BE POSSIBLE FOR A VEHICLE
  4     EITHER ON A LINK UPSTREAM OF SL# OR ON SL# ITSELF (THAT WAS
  4     PREVIOUSLY DELAYED) TO NOW MOVE.>
  3    <BUILD LIST OF POSSIBLE VEHICLES THAT MAY BE ABLE TO NOW MOVE>
  3    LIST:=NULL                     <INITIALIZE LIST TO BE EMPTY>
  3    IF  <PROMPT SL# ITSELF>
  5      FLAG=SELF
  3    THEN
  5      V:=VMEMCH(SLMEMT(SL#))  <SET V TO THE VEH AT HEAD OF SL#>
  5      IF       <V HAS BEEN QUEUED DUE TO CONGESTION/FAILURE>
  7        VQREAS(V)=1
  5      THEN
  7        LIST:=V                      <SET LIST EQUAL TO V>
  5      ENDIF
  3    ELSE   <PROMPT SL(S) UPSTREAM OF SL#>
  5      FOR
  7        EACH SL (SL') IN SLUSLIST(SL#)      <UPSTREAM OF SL#>
  5      DO
  7        IF  <UPSTREAM SL IS NOT A SOURCE>
  9          SL' ¬= SOURCE
  7        THEN
  9          V:=VMEMCH(SLMEMT(SL')) <SET V TO THE VEH AT HEAD OF SL'>
  9          IF  <V HAS BEEN QUEUED DUE TO CONGESTION/FAILURE>
 11            VQREAS(V)=1
  9          THEN
 11            LIST:=LIST+V               <ADD V TO LIST>
  9          ENDIF
  7        ENDIF
  5      ENDDO
  5      REORDER LIST IN FIFO/PRIORITY ORDER
  7        <NOTE THAT JUST THE HEAD VEHICLES OF EACH UPSTREAM SL ARE
  8         INCLUDED IN THIS LIST.  SUBSEQUENT  VEHICLES/TRAINS WILL BE
  5         EXTRACTED BY THE SLPROMPT ISSUED BY THE SLLEAVE OF THE HEAD
  8         VEHICLE/TRAIN, IF THAT HEAD ONE IS, IN FACT, ABLE TO LEAVE>
  3    ENDIF
  5    FOR
  5      EACH VEHICLE(V') ON LIST
  3    DO
  5      RUN SSTEST(V')                  <FIND IF V' CAN MOVE AND, IF SO,
 40                                          INTO WHICH LINK>
  5      IF      <V' CAN ENTER ITS NEXT SL>
  7        AENTRS=0
  5      THEN
  7        VQREAS(V'):=0          <NOTE VEHICLE IS ABOUT TO RETURN TO FEL>
  7        RUN SSLEAV(V')        <LEAVE OLD SL>
  7        VCURR(V'):=AVNXSL     <SET CURRENT SL TO NEXT SL>
  7        IF
  8          AVNXSL=SINK
  7        THEN   <LEAVING THE MODELED AREA>
```

```
      PROJECT: ACT                    MEMBER: SASPRM                    DATE:
      LIBRARY: SIM                    LEVEL:  01.21                     TIME:
      TYPE:    PLL                    USERID: CI26496                   PAGE:
START
CCL   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  4           ACCUMULATE STATISTICS ON TRIPS & VEHICLE
  5           RETURN TRIPS & VEHICLE TXNS TO AVAILABLE CHAIN
  6         ELSE    <NEXT SL IS NOT A SINK>
  7           V.EVNT(V*):=0       <RESET THE VEHICLE EVENT OF V*
  8                                TO NO EVENTS DONE YET>
  9           RUN SLMOD(V*)           <ENTER NEW SL>
 10           <NOTE: THIS USE OF SLMODEL, AS DO ALL OTHER USES EXCEPT
 11                  THE MAIN USE IN SLARCH, ASSUMES THAT SLMODEL WILL
 12                  NEVER SET ANSWER1=1='DONE'>
  7         ENDIF
  8       ENDIF
  9     ENDDO -
  1   ENDPROC
  2     <NOTE: THERE EXISTS A 3RD CASE (OTHER THAN SELF & UPSTREAM) WHERE
 10            IT IS NECESSARY TO RESTART VEHICLES.  IT IS THE CASE WHERE
 11            THE VEHICLE HAD NOT FINISHED PROCESSING ON THE SL IT IS ON
 12            (I.E., NOT 'DONE').  THIS TYPE OF RESTART DONE IN SLLEAVE
 13            BY THE USE OF SLMODEL IMMEDIATELY, SO THAT THE VEHICLE
 14            GOES BACK ON THE FEL IMMEDIATELY.  (THIS IS
 15            VQREAS=3.)    WHEN THE VEHICLE IS 'DONE' (VQREAS=2)
 16            CLPROMPT(_,SELF) IS USED.>
```

```
      PROJECT: ACT                        MEMBER: SATORG                        DATE:
      LIBRARY: DSM                        LEVEL:  01.02                         TIME:
      TYPE:    MIL                        USERID: C120496                       PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--

    1  PROC:SATORG                    <TRIP ARRIVES AT STATION>
    3    IF        <CANNOT FIT>
    5      UCAP(I) < UOCC(I)+TPASS(T*)
    3    THEN
    5      RECORD REJECTION
    3    ELSE        <CAN FIT>
    5      SPLIT TRIP INTO SUBTRIPS USING TSPLIT
    5      FOR
    7        EACH SUBTRIP
    5      DO
    7        GET FREE XTN(I*)
    7        INITIALIZE T*
    7        RUN SUMOD(T*)
    5      ENDDO
    3    ENDIF
    1  ENDPROC
```

```
     PROJECT: AGT                        MEMBER: SATRD
     LIBRARY: DSM                        LEVEL:  01.01
     TYPE:    PDL                        USERID: C120496
SART
COL  ----+----1----+----2----+----3----+----4----+----5----+----
     PROC:SATRD                    <READ TRIP RECORD FROM TAPE>                000100
       GET TRIP TRANSACTION                                                    000200
       DO                                                                      000300
5        READ TORIG,TDEST,TPASS,& TIME INTO TRIP TRANSACTION                   000400
3      UNTIL                                                                   000500
5        TORIG=STSIM                                                           000600
3      ENDDO                                                                   000700
3      IF                                                                      000800
5        END OF FILE                                                           000900
3      THEN                                                                    001000
5        FREE ACTIVE SYSTEM TRANSACTION                                        001100
5        FREE TRIP TRANSACTION JUST ACQUIRED                                   001200
3      ENDIF                                                                   001300
1    ENDPROC                                                                   001400
```

```
      PROJECT: ACT                    MEMBER: SAUCTL                    DATE:
      LIBRARY: USM                    LEVEL:  01.19                     TIME:
      TYPE:    PCL                    USERID: P326507                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--

    1  PROC:SAUCTL(TACTIV)  <TRIP LINK CONTROL--RUN AT TRIP EVENT TIME>
    2    RUN SUMOD(TACTIV)                    <PUT T THROUGH MODEL OF TL>
    3    IF
    4      ADONET = 1
    3    THEN
    5      RUN SUTEST(TACTIV)                 <FIND NEXT TL & TEST IF CAN ENTER>
    5      IF
    7        ACNTRT=1                         <CAN_ENTER>
    6      THEN
    7        IF                               <THERE ARE MORE TRIP LINKS>
    9          AINXTL < 4
    7        THEN
    9          RUN SULEAV(TACTIV)             <LEAVE OLD TL>
    9          TCURR(TACTIV) = AINXTL         <SET CURRENT TL TO NEXT TL>
    9          TNEVNT(TACTIV):=0              <RESET TO NO EVENTS PERFORMED YET>
    9          RUN SUMOD(TACTIV)              <ENTER NEW TL>
    7        ELSE
    9          UPDATE STATS FOR TRIP LEAVING THE FEL STATE
    9          RUN SUTABU (TACTIV)            <PUT T IN BOARDING QUEUE AND
   39                                          GET VEHICLE UPSTREAM MOVING IF
   39                                          APPLICABLE>
    7        ENDIF
    8      ENDIF
    3    ENDIF
    3    IF
    5      TCREAS (TACTIV) ~= 0
    3    THEN
    5      ACCUMULATE STATS FOR THE TRIP LEAVING THE FEL STATE
    5      ACCUMULATE STATS FOR THE TRIP ENTERING THE QUEUED STATE
    3    ENDIF
    1  ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SAUPRM                    DATE:
      LIBRARY: DSM                    LEVEL:  01.07                     TIME:
      TYPE:    PCL                    USERID: CI20496                   PAGE:
STAT
 CDE  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-
      *
    1 PROC: SAUPRM(TL#)   <ALWAYS UPSTREAM>
    2   CASEENTRY(TL#)

    3   CASE(TK)
    4     <NO OP>   <UPSTREAM OF THE TICKETING TL IS THE TRIP SOURCE>

    3   CASE(TS)
    5     T*:=UNEMTL(TK)
    6     IF   <HEAD OF TICKETING TL IS QUEUED DUE TO CONG./FAIL>
    7       TGREAS(T*)=1
    8     THEN
    7       RUN SUTEST(T*)
    7       IF
    9         CAN_ENTER
    7       THEN
    9         RUN SULEAV(T*)
    9         TCURR(T*):=TP
    9         TEVENT(T*):=0
    9         RUN SUMOD(T*)
    7       ENDIF
    5     ENDIF

    5     CASE(SQ)
    3     T*:=UNEMTL(TS)
    6     IF   <HEAD OF TURNSTILE TL IS QUEUED DUE TO CONG./FAIL>
    7       TGREAS(T*)=1
    8     THEN
    7       RUN SUTEST(T*)
    7       IF
    9         CAN_ENTER
    7       THEN
    9         RUN SULEAV(T*)
    9         TCURR(T*):=SQ
    9         TEVENT(T*):=0
    9         RUN SUMOD(T*)
    7       ENDIF

    9     ENDCASE

    3   ENDPROC
    3   <NOTE: THERE EXISTS A 2ND CASE (OTHER THAN UPSTREAM) WHERE
   10          IT IS NECESSARY TO RESTART TRIPS.  IT IS THE CASE WHERE
   10          THE TRIP HAD NOT FINISHED PROCESSING ON THE TL IT IS ON
   10          (I.E., NOT 'DONE').  THIS TYPE OF RESTART DONE IN SULEAV
   10          BY USE OF SUMOD SO THAT THE TRIP GOES BACK ON THE FEL
   10          THIS IS TGREAS=3.    THERE DOES NOT EXIST A CASE
   10          CORRESPONDING TO TGREAS=2 (I.E., TRIP IS 'DONE' BUT
   10          CANNOT LEAVE BECAUSE ANOTHER TRIP IS IN FRONT OF IT)
   10          BECAUSE ONLY ONE TRIP CAN BE IN PROCESSING AT A TIME.>
```

```
      PROJECT: AGT                          MEMBER: SAVORG                        DATE:
      LIBRARY: USR                          LEVEL:  01.03                         TIME:
      TYPE:    PCL                          USERID: C120496                       PAGE:
START
 COL  ----+-----1----+-----2----+-----3----+-----4----+-----5----+-----6----+-----7-

    1  PROC:SAVORG                        <VEHICLE ARRIVES AT THE STATION>
    2    INITIALIZE V#
    3    READ V#'S TRIP FOLLOWER RECORDS
    4    IF
    5      V#_IS HEAD OF TRAIN
    6    THEN
    7      READ VEHICLE RECORDS & THEIR TRIP FOLLOWER RECORDS
    8    ENDIF
    9    RUN SUMCD(V#)
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SAVRD                    DATE:
      LIBRARY: USR                    LEVEL:  01.00                    TIME:
      TYPE:    PLI                    USERID: CI20490                  PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SAVRD                    <READ VEHICLE HEADER RECORD>
    2    GET VEH TRANSACTION
    3    READ TIME, NEXT STOP, DIVERT-TO-STORE,SINK,ROUTE,#PASS.,TRAIN LEN,
    4      SKIP FOLLOWER RECORDS
    5    IF
    6      END OF FILE
    7    THEN
    8      FREE ACTIVE SYSTEM TRANSACTION
    9      FREE TRIP TRANSACTION JUST ACQUIRED
   10    ENDIF
   11  ENDPROC
```

```
    PROJECT: ACT                    MEMBER: SAMTIX                    DATE:
    LIBRARY: DSM                    LEVEL:  01.00                     TIME:
    TYPE:    PDL                    USERID: C120496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC SAMTIX <WRITE INDEX FILE UPDATES>
   3    &SYSDAYTIM TO GET DATE AND TIME
   4    IF
   5      ACKPTI>0
   6    THEN
   5      WRITE LINE TO INDEX FILE INCLUDING MEMBER NAME,DATE, & TIME
   8    ENDIF
   9    IF
   5      ASAMPI>0
   6    THEN
   5      WRITE LINE TO INDEX FILE INCLUDING MEMBER NAME,DATE, & TIME
   8    ENDIF
   1  ENDPROC
```

```
      PROJECT: AGI                    MEMBER: SA2NIT                    DATE:
      LIBRARY: DEM                    LEVEL:  01.01                     TIME:
      TYPE:    PDL                    USERID: C12049B                   PAGE:
STA T
 CCC  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SA2NIT    <INITIALIZE STATISTICS>
    2    <ZERO STATUS STATISTICS
    3    <NON-ZERO    <RESET HISTORICAL STATISTICS>
    4    <WRITE FIRST HEADER RECORD TO SA? STATISTICS FILE DEFINING SIZES
    5      OF DATA TO CCTE
    6  ENDPROC
```

```
      PROJECT: AGI                    MEMBER: SERROR                    DATE:
      LIBRARY: SSM                    LEVEL:  01.00                     TIME:
      TYPE:    PLL                    USERID: C120496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC ERROR <WRITE ERROR MESSAGE & CONTINUE/TERMINATE>
    2    DETERMINE LENGTH OF MESSAGE(LOOK FOR SEMICOLIN)
    3    PRINT MESSAGE TEXT
    4    PRINT STANDARD LINE ASSOCIATED WITH SEVERITY
    5    PRINT VALUE OF CLOCK
    6    INCREMENT NUMBER OF MESSAGES COUNTERS
    7    IF
    8      MESSAGE TYPE IS SEVERE                                    OR
    9      NUMBER OF INFORMATIVE MESSAGES > KMMSGI                   OR
   10      NUMBER OF WARNING MESSAGES > KMMSGW                       OR
   11      NUMBER OF INFORMATIVE + WARNING MESSAGES > KMMSGS         OR
   12      NUMBER OF MESSAGES OF ANY GIVEN ID > KMMTYP
   13    THEN
   14      TERMINATE SIMULATION
   15    ENDIF
    1  ENDPROC
```

```
    PROJECT: AGT                        MEMBER: SMIST
    LIBRARY: SAMPLE                     LEVEL:  00.02
    TYPE:    PDL                        USERID: C120496
LAST
CO'  ----+----1----+----2----+----3----+----4----+----5----+----

    PROC: SMIST <GENERATE HISTOGRAM>                                    000100
  3   COMPUTE MEAN AND VARIANCE OF VARIABLE, AND BIN AMPLITUDE PER MARKER 000200
  3   ACCUMULATE HISTOGRAM                                              000300
  3   PRINT MEAN AND VARIANCE OF VARIABLE, AND BIN AMPLITUDE PER MARKER  000400
  3   PRINT HISTOGRAM                                                   000500
  1 ENDPROC                                                             000600
```

```
      PROJECT: AGT                        MEMBER: SIADDR                        DATE
      LIBRARY: ESM                        LEVEL:  01.01                         TIME
      TYPE:    PDL                        USERID: D425537                       PAGE
START
 COL  ----+-----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  C/*************************************************************************
    1  C/*                                                                       *
    1  C/* MODULE  NAME - SIADDR                                                 *
    1  C/*                                                                       *
    1  C/* FUNCTION: - SIADDR ESTABLISHES THE LOCATION AND LENGTH OF THE         *
    1  C/*             SYSTEM CHARACTERISTICS.                                   *
    1  C/*                                                                       *

    2    PROC=SIADDR:

    4      CALL SISADD( IN SIBWRT) TO SAVE SYSTEM CHARACTERISTICS LOCATION
    9           AND LENGTH

    2    ENDPROC
```

```
    PROJECT: AGT                    MEMBER: SIBWRT
    LIBRARY: DSM                    LEVEL:  01.03
    TYPE:    PDL                    USERID: C120496
CARD
COL ----+----1----+----2----+----3----+----4----+----5----+---
    C/*************************************************************** */ 000100
    C/*                                                               */ 000200
  1 C/* MODULE NAME -  SIBWRT                                         */ 000300
  1 C/*                                                               */ 000400
  1 C/* FUNCTION - ENTRY SISADD SAVES THE SYSTEM CHARACTERISTICS      */ 000500
  1 C/*                         LOCATION AND ADDRESS                  */ 000600
  1 C/*            ENTRY SISWRT WRITES THE SYSTEM CHARACTERISTICS      */ 000700
  1 C/*                         STRUCTURED OUTPUT DATA SET            */ 000800
                                                                       000900
  2  PROC SIBWRT:                                                      001000
                                                                       001100
  4    ENTRY SISADD:  <INITIALIZATION ENTRY>                           001200
                                                                       001300
  6      ESTABLISH ADDRESS AND LENGTH OF SYSTEM CHARACTERISTICS        001400
 11           STRUCTURED DATA AREA                                     001500
                                                                       001600
  4    RETURN                                                          001700
                                                                       001800
  4    ENTRY SISWRT:  <DATA WRITE ENTRY>                               001900
                                                                       002000
  6      WRITE SYSTEM CHARACTERISTICS STRUCTURED DATA FILE             002100
                                                                       002200
  4    RETURN                                                          002300
                                                                       002400
  2  ENDPROC                                                           002500
```

```
      PROJECT: ACT                    MEMBER: SICHCK                    DATE
      LIBRARY: DSM                    LEVEL:  01.02                     TIME
      TYPE:    PDL                    USERID: J425537                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  C/*************************************************************************
    1  C/*                                                                       *
    1  C/* MODULE  NAME - SICHCK                                                 *
    1  C/*                                                                       *
    1  C/* FUNCTION - SICHCK DOES PARAMETER CHECKING AND INITIALIZATION          *
    1  C/*              INCLUDED SEGMENTS - SICHCK1                              *
    1  C/*                                                                       *

    2   PROC SICHCK:

    4     CONVERT DATA SAMPLING AND
   12           CHECKPOINT INTERVALS FROM SECONDS TO CLOCK UNITS.

    4     CHECK VEHICLE CAPACITY AND
   10           TRIP SPLIT SIZE FOR ERRORS

    4     DO FOR EACH STATION LINK

    6       CONVERT
   14               LINK TRAVEL TIME,
   14               HEADWAY TRAVEL TIME
   34                                   FROM SECONDS TO CLOCK UNITS.

    4     ENDDO

    4     DO FOR EACH ROUTE

    6       INITIALIZE
   17                 'NEXT TO LEAVE',
   16                  VEHICLE SPACING,
   17                 'LAST TO LEAVE'
   33                                   IN CLOCK UNITS

    4     ENDDO

    4     DO FOR EACH ENTRY IN MERGE DELAY TIME DISTRIBUTION

    6       CONVERT TIME FROM SECONDS TO CLOCK UNITS

    4     ENDDO

    4     DO FOR EACH ENTRY IN EMPTY VEHICLE DELAY TIME DISTRIBUTION

    6       CONVERT TIME FROM SECONDS TO CLOCK UNITS

    4     ENDDO

    4     CONVERT DEBOARD/BOARD TIMES FROM SECONDS TO CLOCK UNITS

    4     DO FOR EACH TRIP LINK

    6       CONVERT TRAVEL TIMES FROM SECONDS TO CLOCK UNITS

    4     ENDDO

    4     ESTABLISH POINTER TO STATION LINK TYPES
    4     DOUNTIL STORAGE LINK IS FOUND
    6       INCREMENT POINTER
    4     ENDDO
    4     SAVE STORAGE LINK ID

    4     INCLUDE SICHCK1 TO DO LINK-EVENT COMPATIBILITY CHECKS

    2   ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SICHCK1                    DATE
      LIBRARY: DSM                    LEVEL:  01.00                      TIME
      TYPE:    PLI                    USERID: D425537                    PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1 C/*************************************************************** *
    1 C/*                                                               *
    1 C/* SEGMENT NAME - SICHCK1                                        *
    1 C/*                                                               *
    1 C/* FUNCTION - SICHCK1 IS INCLUDED BY DSM PARAMETER CHECKING AND  *
    1 C/*              INIT. TO DO LINK-EVENT COMPATIBILITY CHECKING.   *
    1 C/*                                                               *
    4    DO FOR EACH LINK

    6      DO WHILE A VALID LINK EVENT TYPE EXISTS FOR THIS LINK

    9        IF THIS EVENT IS BOARD,
   25                         DEBOARD, OR
   25                         DEBOARD/BOARD THEN
   10          SET INDICATOR FOR BOARD/DEBOARD

    6        ELSE

   10          IF THIS IS A STORAGE OR LAUNCH EVENT AND
   13             IT IS NOT THE LAST EVENT ON THE LINK THEN

   12            SET ERROR INDICATOR

   10          ELSE

   12            IF THIS IS THE LAST EVENT ON A STORAGE LINK AND
   15               IT IS NOT A STORAGE EVENT THEN

   14              SET ERROR INDICATOR

   13              END IF

   10          END IF

    3        END IF

    5        INCREMENT EVENT POINTER

    6      ENDDO

    8      IF A DEBOARD/BOARD EVENT AND DOWNSTREAM STATION LINK
   11           HAVE NOT OCCURRED TOGETHER
   38                                      THEN
    9          SET ERROR INDICATOR

    6      END IF

    4    ENDDO

    4    IF ANY ERRORS FOUND ABOVE THEN

    6      CALL SINERR TO WRITE ERROR MESSAGE

    4    END IF

    4    DO FOR EACH LINK

    8      INITIALIZE UPSTREAM LINK POINTER
    8      COUNT IF A SOURCE LINK IS FOUND

    8        IF CURRENT LINK IS A SOURCE THEN

   10          SAVE THE LINK ID FOR MODEL

    8        END IF
    5        INCREMENT UPSTREAM LINK POINTER

    8      ENDDO

    4    ENDDO
```

```
      PROJECT: ACT                    MEMBER: SICUMP                    DATE
      LIBRARY: DSM                    LEVEL:  01.05                     TIME
      TYPE:    PDL                    USERID: D425537                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  C/*************************************************************************
    1  C/*                                                                       *
    1  C/* MODULE  NAME - SICUMP                                                 *
    1  C/*                                                                       *
    1  C/* FUNCTION - SICUMP CONVERTS A FREQUENCY DISTRIBUTION TO                 *
    1  C/*              CUMULATIVE PROBABILITY DISTRIBUTION.                     *
    1  C/*                                                                       *
    2   PROC SICUMP:

    4     IF
    6       FIRST ENTRY < 0
    4     THEN

    6       ERROR
    4     ENDIF

    4     INDEX:= 1
    4     WHILE
    6       ENTRY(INDEX) > 0 AND NO ERRORS
    4     DO

    6       IF
    8         ENTRY(INDEX) < 0
    6       THEN

    6         ERROR - NEGATIVE PROBABILITY VALUE

    6       ELSE

    8         IF
   10           DISTRIBUTION MEAN REQUESTED
    8         THEN

   10           MEAN = MEAN + ENTRY(INDEX) * INDEX

    8         ENDIF

    8         ENTRY(INDEX) = ENTRY(INDEX) + ENTRY(INDEX-1)

    8         IF
   10           ENTRY(INDEX) > 1.0
    8         THEN

   10           ERROR - PROBABILITY > 1.0
    8         ENDIF

    8         INDEX = INDEX + 1
    6       ENDIF

    4     ENDDO

    4     IF
    6       ENTRY(INDEX) < .99      OR
    6       ENTRY(INDEX) > 1.0
    4     THEN

    6       ERROR - LAST VALUE MUST BE 1.0
    4     ENDIF

    2   ENDPROC SICUMP
```

```
      PROJECT: AGT                          MEMBER: SIGIAT                          DATE
      LIBRARY: DSM                          LEVEL:  01.04                           TIME
      TYPE:    PCL                          USERID: D425537                         PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  C/*************************************************************************
    1  C/*                                                                       *
    1  C/* MODULE  NAME - SIGIAT                                                 *
    1  C/*                                                                       *
    1  C/* FUNCTION -  SIGIAT DETERMINES THE NEXT INTERARRIVAL TIME (IAT)FOR     *
    1  C/*             DSM VEHICLE GENERATION                                    *
    1  C/*                                                                       *
    2   PROC SIGIAT:

    4     IF
    6       USER IAT DISTRIBUTION OPTION  <DVIASW>
    4     THEN

    6       CALL SMRSEL TO SAMPLE THE USER'S DISTRIBUTION FOR IAT

    4     ELSE  <USE EXPONENTIAL ARRIVAL RATE>

    6       CALL SMRNG FOR A RANDOM NUMBER

    8       IF
    8         RANDOM <= P(MIN HEADWAY)   <DVPMIN>
    8       THEN

    3         NEXT IAT = HEADWAY
    8         <DVIATM = DVHDWY>

    6       ELSE <COMPUTE INTERARRIVAL TIME>

    9         NEXT IAT = MINIMUM HEADWAY -
   19                    ((MEAN IAT - MINIMUM HEADWAY)/
   19                    (1 - P(MIN HEADWAY)) *
   19                    LOG((1 - RANDOM PROBABILITY)/
   19                    (1 - P(MIN HEADWAY))
    8         <DVIATM = DVHDWY -
   18                   ((DVLMDA - DVPMIN)/(1 - DVPMIN)) *
   15                   LOG((1 - DRANDO)/(1 - DVPMIN))>

    4     ENDIF

    4     IF
    6       SYNCHRONOUS ENVIRONMENT  <DVSNSW>
    4     THEN

    6       IAT = IAT * SLOT LENGTH IN TIME
    8       <DVIATM = DVIATM * DVSLOT>

    4     ENDIF
    2   ENDPROC SIGIAT
```

```
PROJECT: AGT                    MEMBER: SIINIT                  DATE
LIBRARY: USM                    LEVEL:  01.02                   TIME
TYPE:    PLI                    USERID: D425537                 PAGE
START
COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  C/*** ***************************************************************/
   1  C/*                                                                 */
   1  C/* MODULE  NAME - SIINIT                                           */
   1  C/*                                                                 */
   1  C/* FUNCTION - SIINIT DOES INPUT PROCESSOR INITIALIZATION           */
   1  C/*                                                                 */

   2   SIINIT: PROC;

   4     CALL LOCCOM AND SIADDR TO
   5          ESTABLISH THE ADDRESS AND LENGTH OF THE
   6          SYSTEM CHARACTERISTICS AREA

   4     CALL SIPARM TO DECODE THE INPUT PROCESSOR PARAMETER LIST

   4     ESTABLISH DEFAULTS FOR MODEL PROCESSOR OPTIONS

   4     ESTABLISH DEFAULTS FOR EVENT CONTROL PARAMETERS

   4     ESTABLISH DEFAULTS FOR RUNTIME LIMITS PARAMETERS

   4     DO FOR THE MAXIMUM NUMBER OF STATION LINKS

   5       ESTABLISH DEFAULTS FOR STATION LINK CHARACTERISTICS

   4     ENDDO

   4     CLEAR PROBABILITY DISTRIBUTION ARRAYS FOR
   5          EMPTY VEHICLE DELAY
   6          MERGE DELAY
   7          PASSENGER TRANSFER
   8          PASSENGER-VEHICLE COMPABITILITY
   9          EMPTY VEHICLE NEEDED ELSEWHERE

   4     DO FOR THE MAXIMUM NUMBER OF ROUTES

   5       CLEAR ROUTE LISTS AND POINTERS
   6       CLEAR ROUTE CHARACTERISTICS ARRAYS

   4     ENDDO

   4     ESTABLISH DEFAULTS FOR MISCELLANEOUS PARAMETERS

   4     DO FOR THE MAXIMUM NUMBER OF TRIP LINKS

   5       CLEAR TRIP LINK CHARACTERISTICS ARRAYS

   4     ENDDO

   2   ENDPROC SIINIT
```

```
    PROJECT: AGT                    MEMBER: SIMNAM
    LIBRARY: DSM                    LEVEL:  01.03
    TYPE:    PDL                    USERID: C12049o
.ART
SOL ----+----1----+----2----+----3----+----4----+----5----+--

    C/*************************************************************** */ 000100
    C/*                                                               */ 000200
 1  C/* MODULE NAME -  SIMNAM                                         */ 000300
 1  C/*                                                               */ 000400
 1  C/* FUNCTION -  SIMNAM DECODES THE INPUT PROCESSOR PARAMETER LIST */ 000500
 2   PROC SIMNAM:                                                      000600
                                                                       000700
 4     INITIALIZE OUTPUT ARRAY                                         000800
                                                                       000900
 4     DO FOR FIRST SIX PARAMETER LIST FIELDS                          001000
                                                                       001100
 6       INITIALIZE PARAMETER LIST POINTER                             001200
                                                                       001300
 6       DO WHILE A COMMA (FIELD DELIMITER) IS NOT FOUND               001400
                                                                       001500
 8         SAVE THE CURRENT CHARACTER OF THE PARAMETER LIST            001600
13             IN THE OUTPUT ARRAY                                     001700
                                                                       001800
 8         INCREMENT PARAMETER LIST POINTER                            001900
                                                                       002000
 6       ENDDO                                                         002100
                                                                       002200
 4     ENDDO                                                           002300
                                                                       002400
 4     SAVE THE LAST CHARACTER IN THE PARAMETER LIST AS THE            002500
 9         VEHICLE DEMAND SOURCE INDICATOR                             002600
                                                                       002700
 4     RUN DAYTIM                                                      002800
 4     WRITE LOAD MODULE NEME, DATE & TIME                             002900
 2   ENTRY SIMNAM                                                      003000
 4     LIST ALL USED FILES IN INDEX                                    003100
                                                                       003200
 2   ENDPROC SIMNAM                                                    003300
```

```
    PROJECT: AGT                        MEMBER: SINPUT
    LIBRARY: DSM                        LEVEL:  01.07
    TYPE:    PDL                        USERID: C12049o
ART
OL  ----+----1----+----2----+----3----+----4----+----5----+-
    C/*************************************************************/ 000100
    C/*                                                           */ 000200
 1  C/* MODULE  NAME - SINPUT                                     */ 000300
 1  C/*                                                           */ 000400
 1  C/* FUNCTION -  SINPUT IS THE DSM INPUT PROCESSOR CONTROL ROUTINE */ 000500
 1  C/*             NEXT LEVEL OF INCLUDED SEGMENTS               */ 000600
 1  C/*               SINPUT1 - READ SYSTEM CHARACTERISTICS       */ 000700
 1  C/*               SINPUT2 - PROCESS RUNTIME DATA              */ 000800
 1  C/*               SINPUT3 - READ TRIP DEMAND DATA             */ 000900
 1  C/*               SINPUT4 - READ VEHICLE DEMAND DATA          */ 001000
                                                                    001100
 2   CALL SIINIT TO DO SYSTEM INITIALIZATION                        001200
                                                                    001300
 2   INCLUDE SINPUT1 TO READ SYSTEM CHARACTERISTICS                 001400
                                                                    001500
 2   READ FIRST RUNTIME RECORD                                      001600
                                                                    001700
 2   UNTIL                                                          001800
 4     EOD                                                          001900
 2   DO                                                             002000
                                                                    002100
 4     UNTIL                                                        002200
 6       NEW TIME TAG FOUND                                         002300
 4     DO                                                           002400
                                                                    002500
 6       POINT TO START OF INPUT DATA TYPES <ATYPE>                 002600
                                                                    002700
 6       WHILE                                                      002800
 6         RUNTIME RECORD ¬= CURRENT ATYPE                          002900
 6       DO                                                         003000
                                                                    003100
 8         IF                                                       003200
10           RUNTIME RECORD = CURRENT ATYPE                        003300
 8         THEN                                                     003400
                                                                    003500
             INCLUDE SINPUT2 TO PROCESS RUNTIME DATA                003600
                                                                    003700
 8         ELSE <NO MATCH YET>                                      003800
                                                                    003900
10           IF                                                     004000
12             END OF ATYPE                                         004100
10           THEN                                                   004200
                                                                    004300
12             ERROR CONDITION                                      004400
                                                                    004500
10           ENDIF                                                  004600
                                                                    004700
 8         ENDIF                                                    004800
                                                                    004900
 8         INCREMENT   ATYPE POINTER                                005000
                                                                    005100
 6       ENDDO                                                      005200
                                                                    005300
 6       READ NEXT RUNTIME RECORD                                   005400
                                                                    005500
                                                                    005600
 6       IF                                                         005700
 9         RUNTIME TIME TAG IS ZERO                                 005800
 6       THEN                                                       005900
                                                                    006000
 8         IF                                                       006100
10           TRIP GEN REQUESTED                                     006200
 8         THEN                                                     006300
                                                                    006400
10           INCLUDE SINPUT3 TO READ TRIP DEMAND DATA               006500
                                                                    006600
10           CALL SITDGN TO GENERATE TRIPS                          006700
                                                                    006800
1            WRITE RUN INDEX DATA FOR TRIP FILE                     006900
                                                                    007000
```

```
    PROJECT: AGT                    MEMBER: SINPUT
    LIBRARY: DSM                    LEVEL:  01.07
    TYPE:    PDL                    USERID: C120496
START
COL ----+----1----+----2----+----3----+----4----+----5----+·

        ENDIF                                                                  007100
                                                                               007200
        IF                                                                     007300
10        VEH GEN REQUESTED                                                    007400
 8      THEN                                                                   007500
                                                                               007600
10        INCLUDE SINPUT4 TO READ VEHICLE DEMAND DATA                          007700
                                                                               007800
10        CALL SIVDGN TO GENERATE VEHICLES                                     007900
                                                                               008000
10        WRITE RUN INDEX DATA FOR VEHICLE FILE                                003100
                                                                               008200
 8      ENDIF                                                                  008300
                                                                               008400
 6      IF                                                                     008500
11        MODEL SETUP  REQUESTED                                               008600
 8      THEN                                                                   008700
                                                                               008800
10        IF                                                                   008900
12          STATION CONFIGURATION REQUESTED                                    009000
10        THEN                                                                 009100
                                                                               009200
12          CALL SISCFG TO CONFIGURE STATION LINKS                             009300
                                                                               009400
10        ENDIF                                                                009500
                                                                               009600
10        CALL SICHCK TO DO PARAMETER CHECKING AND INITIALIZATION              009700
                                                                               009800·
10        CALL SIREPT TO WRITE INITIAL CONDITIONS REPORT                       009900
                                                                               010000
10        CALL SISWRT TO WRITE STRUCTRD DATA SYSTEM CHARACTERISTICS FILE       010100
                                                                               010200
 8      ENDIF                                                                  010300
                                                                               010400
      ENDIF                                                                    010500
                                                                               010600
    ENDDO                                                                      010700
                                                                               010800
 2  ENDDO                                                                      010900
                                                                               011000
 2  RUN SIWNAM(EP-SIMNAM) TO LIST IN INDEX MEMBERS USED                        011100
                                                                               011200
 2  ENDPROC                                                                    011300
```

```
      PROJECT: ACT                        MEMBER: SINPUT1                        DATE
      LIBRARY: DSM                        LEVEL:  01.03                          TIME
      TYPE:    PDL                        USERID: D425537                        PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  C/*****************************************************************
    1  C/*                                                               *
    1  C/* SEGMENT NAME - SINPUT1                                        *
    1  C/*                                                               *
    1  C/* FUNCTION - SINPUT1 IS INCLUDED BY SINPUT TO READ THE SYSTEM   *
    1  C/*            CHARACTERISTICS FILE                               *
    1  C/*                                                               *

    2   UNTIL
    4     END OF SYSTEM CHARACTERISTICS
    2   DO

    4     READ RECORD

    4     VERIFY ZERO TIME TAG

    4     POINT TO INPUT DATA TYPES LIST (ATYPE)

    4     WHILE
    6       SYSTEM CHARACTERISTICS TYPE ¬= CURRENT ATYPE
    4     DO
    6       IF
    9         SYSTEM CHATACTERISTICS INPUT TYPE = CURRENT ATYPE
    6       THEN

    8         PROCESS BY TYPE

   10           TYPE:
   10           - CHECKPOINT
   10           - RESTART
   10           - STOP
   10           - END OF DATA
   10           - INDEX
   10           - FAILURE/REPAIR
   10           - AUXILIARY OUTPUT
   10           - TRIP
   10           - VEHICLE

   14               ALL ARE ILLEGAL TYPES IN SYSTEM CHARACTERISTICS INPUT


   10           TYPE:
   10           - TEXT

   14               WRITE DATA TO OUTPUT FILE

   10           TYPE:
   10           - PARAMETER
   10           - OPTION
   10           - SELECT
   10           - DATA

   14               CALL NDBOR TO READ INPUT


   10           TYPE:
   10           - COMMENT

   14               IGNORE AND  ADVANCE TO NEXT INPUT

    6       ELSE   <INPUT TYPE DOES NOT MATCH VALID SYSCHAR TYPE>

    8         IF
   10           END OF ATYPE
    8         THEN

   10           ERROR CONDITION - INVALID ENTRY TYPE

    8         END IF
```

```
        PROJECT: AGT                    MEMBER: SINPUT1                 DATE:
        LIBRARY: DSN                    LEVEL:  01.03                   TIME:
        TYPE:    PLL                    USERID: L426537                 PAGE:
START
COL    ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  3         .NDIF

  4       ENDDO

  2     ENDDO
```

```
      PROJECT: AGT                    MEMBER: SINPUT2                    DATE
      LIBRARY: DSM                    LEVEL:  01.02                      TIME
      TYPE:    PDL                    USERID: D425537                    PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1 C/*************************************************************** *,
    1 C/*                                                                *,
    1 C/* SEGMENT NAME - SINPUT2                                         *,
    1 C/*                                                                *,
    1 C/* FUNCTION - SINPUT2 IS INCLUDED BY SINPUT  TO PROCESS RUNTIME   *,
    1 C/*              DATA                                              *,
    1 C/*                                                                *,

   10         -   PROCESS RUNTIME DATA BY TYPE                           *,

   12               TYPE
   12               - CHECKPOINT
   12               - STOP

   16                    WRITE DATA TO MODEL RUNTIME FILE


   12               TYPE
   12               - RESTART

   16                    NOT A LEGAL TYPE HERE


   12               TYPE
   12               - INDEX

   16                    WRITE INDEX DATA TO RUN INDEX FILE


   12               TYPE
   12               - TEXT

   16                    WRITE TO MODEL RUNTIME FILE


   12               TYPE
   12               - PARAMETER
   12               - OPTION
   12               - SELECT
   12               - DATA

   16                    IF
   16                      ZERO TIME ENTRY
   16                    THEN

   16                      CALL NOBOR TO READ DATA

   16                    ELSE  <NON ZERO INPUT>

   16                      WRITE DATA TO MODEL RUNTIME FILE

   16                    ENDIF


   12               TYPE
   12               - FAILURE/REPAIR
   12               - TRIP
   12               - VEHICLE

   16                    WRITE DATA TO MODEL RUNTIME FILE


   12               TYPE
   12               - AUXILIARY OUTPUT

   16                    IF THIS INPUT IS IN ZERO TIME THEN

   16                      CALL  SAFLAG TO PROCESS DATA

   16                    ELSE  <NON ZERO TIME INPUT>
```

```
START
 COL ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  14                    WRITE DATA TO MODEL RUNTIME FILE

  16                  ENDIF

  12              TYPE
  12              = COMMENT

  16                    IGNORE AND ADVANCE TO NEXT INPUT

  10            END OF RUNTIME DATA TYPES TO BE PROCESSED
```

```
   PROJECT: AGT                      MEMBER: SINPUT3
   LIBRARY: DSM                      LEVEL:  01.02
   TYPE:    PDL                      USERID: C120496
:ART
:DL  ----+----1----+----2----+----3----+----4----+----5----+---
     C/*************************************************************** 000100
     C/*                                                             */ 000200
     C/* SEGMENT NAME - SINPUT3                                      */ 000300
   1 C/*                                                             */ 000400
   1 C/* FUNCTION - SINPUT2 IS INCLUDED BY DSM INPUT TO READ TRIP DEMAND */ 000500
   1 C/*            DATA INTO COMMON SCNTDM                          */ 000600
   1 C/*                                                             */ 000700
  13             READ                                                  000800
  18                  STATION ID AND - DTSTAN                          000900
  18                  START TRIP GEN TIME (SECONDS)- DTVARS            001000
  18                  END TRIP GEN TIME (SECONDS)- DTENDT              001100
                                                                       001200
  13             READ                                                  001300
  18                  INTERARRIVAL TIME OPTION - DTIASW                001400
                                                                       001500
  13             IF                                                    001600
  15               IAT OPTION = 'USER'                                 001700
  13             THEN                                                  001800
                                                                       001900
  15               READ AND VERIFY                                     002000
  20                    NUMBER OF IAT DIST. ENTRIES - KNIAT            002100
  15               READ                                                002200
  20                    USER IAT DISTRIBUTION - DTIATD(KNIAT)          002300
                                                                       002400
  13             ELSE                                                  002500
                                                                       002600
  15               READ                                                002700
  20                    MEAN INTERARRIVAL TIME -DTLMDA                 002800
                                                                       002900
  13             ENDIF                                                 003000
                                                                       003100
  13             READ AND VERIFY                                       003200
  18                  NO. ENTRIES IN DESTINATION DIST. - KNS           003300
                                                                       003400
  13             READ                                                  003500
                      DESTINATION DISTRIBUTION - DTDEST(KNS)           003600
                                                                       003700
  13             READ AND VERIFY                                       003800
  18                  MAXIMUM TRIP SIZE VALUE - KNNP                   003900
                                                                       004000
  13             READ                                                  004100
  18                  TRIP SIZE DISTRIBUTION - DTPASD(KNNP)            004200
                                                                       004300
   1 C/*********************************END OF SINPUT3**************** */ 004400
```

```
    PROJECT: AGT                    MEMBER: SINPUT4
    LIBRARY: DSM                    LEVEL:  01.0e
    TYPE:    PDL                    USERID: C120496
ART
.OL  ----+----1----+----2----+----3----+----4----+----5----+----6--

     C/*************************************************************** */ 000100
     C/*                                                               */ 000200
     C/* SEGMENT NAME - SINPUT4                                        */ 000300
   1 C/*                                                               */ 000400
   1 C/* FUNCTION - SINPUT4 IS INCLUDED BY DSM INPUT TO READ VEHICLE   */ 000500
   1 C/*            DEMAND DATA INTO COMMON SCNVDM                     */ 000600
   1 C/*                                                               */ 000700
  13           READ                                                    */ 000800
  18                STATION ID - DVSTAN                                  000900
  18                SERVICE POLICY INDICATOR - DVSERV                    001000
  18                START TIME OF GENERATION (SECS) -DVARVS              001100
  18                END TIME OF GENERATION (SECS) -DVENDT                001200
                                                                         001300
  13           READ                                                      001400
  18                VEH CAPACITY - DVCAPP                                001500
  13           READ                                                      001600
  18                SYNCH ENVIRONMENT INDICATOR - DVSNSW                 001700
                                                                         001800
  13           IF                                                        001900
  15             ENVIRONMENT IS SYNCHRONOUS                              002000
  13           THEN                                                      002100
  15             READ                                                    002200
  20                  SLOT LENGTH (SECS) - DVSLOT                        002300
                                                                         002400
  13           ENDIF                                                     002500
                                                                         002600
  13           INCLUDE SINPUT4A TO READ NEXT STOP SELECTION DATA         002700
                                                                         002800
  13           INCLUDE SINPUT4B TO READ INTERARRIVAL TIME DATA           002900
                                                                         003000
  13           READ                                                      003100
  18                SINK DISTRIBUTION - DVSNKD(3)                        003200
                                                                         003300
  13           READ AND VERIFY                                           003400
                    MAX TRAIN LENGTH -KNTLEN                             003500
                                                                         003600
               IF                                                        003700
  15             SERVICE POL = SCHEDULED                                 003800
  13           THEN                                                      003900
                                                                         004000
                                                                         004100
  15             READ                                                    004200
  20                  TRAIN LENGTH FOR EACH ROUTE - DVTLNR(KNR)          004300
                                                                         004400
  13           ELSE  <DEMAND RESPONSIVE>                                 004500
                                                                         004600
  15             READ                                                    004700
  21                  TRAIN LENGTH DISTRIBUTION - DVTLND(KNTLEN)         004800
                                                                         004900
  13           ENDIF                                                     005000
                                                                         005100
  13           IF                                                        005200
  15             SERVICE ¬= DEMAND RESP. SINGLE                          005300
  13           THEN                                                      005400
                                                                         005500
                                                                         005600
  15             READ AND VERIFY                                         005700
  21                  NUMBER OF DESTINATION STATIONS - KNS               005800
  15             READ                                                    005900
  21                  DESTINATION DISTRIBUTION - DVDESD(KNS)             006000
                                                                         006100
  13           ENDIF                                                     006200
                                                                         006300
  13           INCLUDE SINPUT4C TO READ TRIP SIZE DATA                   006400
   1 C/*************************END OF SINPUT4*************************006500
```

```
PROJECT: AGT                    MEMBER: SINPUT4A                    DATE
LIBRARY: DSM                    LEVEL:  01.03                       TIME
TYPE:    PPL                    USERID: 5426537                     PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  C/*************************************************************
    1  C/*                                                           *
    1  C/* SEGMENT NAME - SINPUT4A                                   *
    1  C/*                                                           *
    1  C/* FUNCTION - SINPUT4A IS INCLUDED BY DSM VEH INPUT TO READ NEXT *
    1  C/*              STOP DATA INTO COMMON SCNVDM                 *
    1  C/*                                                           *

   13               IF
   15                 SERVICE = SCHEDULED
   15               THEN

   15                 READ
   20                      NUMBER OF ROUTES - KNR
   20                      METHOD OF CHOOSING NEXT STOP - DVNXID

   15                 VERIFY NUMBER OF ROUTES

   15                 IF
   17                   NEXT STOP CHOSEN FROM ROUTE LIST
   15                 THEN

   17                   FOR
   19                     ROUTES 1 TO KNR
   17                   DO

   19                     READ AND VERIFY
   24                          NUMBER OF STATIONS ON THE ROUTE -DVRELS

   19                     READ
   24                          THE STATIONS ON THE ROUTE - DVRSCH(DVRELS)

   17                   ENDLO

   15                 ELSE   <METHOD IS ROUTE INDICATOR>

   17                   READ
   22                        NEXT STOP INDICATOR - DVRTST(KNR)

   15                 ENDIF

   15               ELSE    <DEMAND RESPONSIVE SERVICE>

   15                 READ AND VERIFY
   20                      P(STOPPING AT SIMPD STATION) - DVSTPP

   13               ENDIF

    1  C/*************************END OF SINPUT4A*************************
```

```
      PROJECT: ACT                    MEMBER: SINPUT4B                    DATE
      LIBRARY: DSM                    LEVEL:  01.03                       TIME
      TYPE:    PL1                    USERID: 0425537                     PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  C/*****************************************************************
    1  C/*                                                               *
    1  C/* SEGMENT NAME - SINPUT4B                                       *
    1  C/*                                                               *
    1  C/* FUNCTION - SINPUT4B IS INCLUDED BY DSM VEH INPUT TO READ      *
    1  C/*              INTERARRIVAL TIME (IAT) DATA INTO COMMON SCNVDM   *
    1  C/*                                                               *

   13               IF
   15                 IAT OPTION = 'USER'
   13               THEN

   15                 READ AND VERIFY
   20                      NO. ENTRIES IN USER DISTRIBUTION - KNIAV

   15                 READ
   20                      USER IAT DIST. - DVIATD(KNIAV,2)

   13               ELSE     < EXPONENTIAL INTERARRIVAL TIME DISTRIBUTION>

   15                 READ
   20                      MEAN INTERARRIVAL TIME - DVLMDA

   15                 READ
   20                      MINIMUM HEADWAY (SECONDS) - DVHDWY
   20                      P(MIN HEADWAY) - DVPMIN

   13               ENDIF

    1  C/*************************END OF SINPUT4B*************************/
```

```
PROJECT: ACT                    MEMBER: SINPUT4C                    DATE
LIBRARY: DSM                    LEVEL:  01.01                       TIME
TYPE:    PLC                    USERID: 0425537                     PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1   C/*************************************************************
    1   C/*                                                           *
    1   C/* SEGMENT NAME - SINPUT4C                                   *
    1   C/*                                                           *
    1   C/* FUNCTION - SINPUT4C IS INCLUDED BY DSM VEH INPUT TO READ TRIP *
    1   C/*            SIZE DATA INTO COMMON SCNVDM                   *
    1   C/*                                                           *

   16       -          READ AND VERIFY
   21                      MAXIMUM TRIP SIZE - KNNP

   16                  READ
   21                      TRIP SIZE DISTRIBUTION - DVPASD(KNNP)

   16                  READ AND VERIFY
   21                      MAXIMUM NUMBER OF TRIPS/VEHICLE - KNNT

   16                  READ
   21                      TRIPS/VEHICLE DISTRIBUTION - DVTRPD(KNNT)

    1   C/*************************END OF SINPUT4C*************************
```

```
      PROJECT: AGT                    MEMBER: SIPARM                    DATE:
      LIBRARY: DSM                    LEVEL:  01.01                     TIME:
      TYPE:    PCL                    USERID: 0425537                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1 C/*************************************************************** */
    1 C/*                                                               */
    1 C/* MODULE  NAME -  SIPARM                                        */
    1 C/*                                                               */
    1 C/* FUNCTION - ENTRY SIPARM SAVES THE ADDRESS OF THE INPUT PARAMETER */
    1 C/*                          LIST                                 */
    1 C/*              ENTRY SIPLST CONTROLS THE DECODING OF THE PARAMETER */
    1 C/*                    LIST                                       */

    2   PROC SIPARM:  <PROCESS DSM INPUT PROCESSOR PARAMETER LIST>

    4     ENTRY SIPARM  <SAVE PARAMETER LIST>

    6       SAVE PARAMETER LIST ADDRESS FOR LATER USE

    6       PASS CONTROL TO INPUT PROCESSOR (SINPUT)

    4     RETURN

    4     ENTRY SIPLST  <PROCESS INPUT PROCESSOR PARAMETER LIST>

    6       CALL SIMNAM TO DECODE PARAMETER LIST

    4     RETURN

    2   ENDPROC SIPARM
```

```
      PROJECT: AGT                    MEMBER: SIPSAV                    DATE
      LIBRARY: GSM                    LEVEL:  01.00                     TIME
      TYPE:    PDL                    USERID: D429537                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  C/*************************************************************************/
    1  C/*                                                                       */
    1  C/* MODULE  NAME -  SIPSAV                                                */
    1  C/*                                                                       */
    1  C/* FUNCTION - SIPSAV SAVES THE ADDRESS AND LENGTH OF THE SYSTEM          */
    1  C/*              CHARACTERISTICS COMMONS                                  */
    1  C/*              ENTRY POINTS: SIPSAV                                     */
    1  C/*                            LODCOM                                     */
    3    PROC SIPSAV:

    5      ENTRY SIPSAV
    7        PROVIDE SAVE LOCATIONS
    9          FOR ADDRESS OF INPUT COMMONS
   13              ADDRESS OF SYSTEM CHARACTERISTICS COMMONS
   13              ADDRESS OF END OF SYSTEM CHARACTERISTICS COMMONS
    5      ENTRY LODCOM
    7        RETURN TO CALLER HAVING CAUSED SIPSAV TO BE
    9          LOADED AND ADDRESSES MADE AVAILABLE

    3    PROCEND
    1  C/*****************************END OF SIPSAV*****************************/
```

```
      PROJECT: ACT                    MEMBER: SIREPT                    DATE
      LIBRARY: DSM                    LEVEL:  01.00                     TIME
      TYPE:    PL1                    USERID: D425937                   PAGE
STAMT
COL   ---+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  C/*****************************************************************
   1  C/*                                                               *
   1  C/* MODULE  NAME - SIREPT                                         *
   1  C/*                                                               *
   1  C/* FUNCTION -  SIREPT WRITES THE DSM INITIAL CONDITIONS REPORT   *
   1  C/*               NEXT LEVEL OF INCLUDED SEGMENTS -               *
   1  C/*                 SIREPT3 - TRIP AND VEHICLE DATA               *
   1  C/*                 SIREPT4 - SERVICE POLICY DATA                 *
   1  C/*                 SIREPT5 - STATION LINK SUMMARY                *
   1  C/*                 SIREPT6 - TRIP LINK SUMMARY                   *
   1  C/*                                                               *
   1  C/*                                                               *
   2   PROC SIREPT: < INITIAL CONDITIONS REPORT>

   4     INCLUDE SIREPT3 TO WRITE TRIP AND VEHICLE CHARACTERISTICS

   4     INCLUDE SIREPT4 TO WRITE SERVICE POLICY CHARACTERISTICS

   4     INCLUDE SIREPT5 TO WRITE STATION LINK SUMMARY

   4     INCLUDE SIREPT6 TO WRITE TRIP LINK SUMMARY

   4     IF ANY SERIOUS ERRORS WERE FOUND  THEN

   6       TERMINATE THE RUN

   4     ENDIF

   2   ENDPROC SIREPT
   1  C/*****************END OF SIREPT*****************************/
```

```
PROJECT: AGI                    MEMBER: SIREPT3                 DATE:
LIBRARY: GSM                    LEVEL:  01.00                   TIME:
TYPE:    PLL                    USERID: 0425537                 PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  C/*************************************************************/
    1  C/*                                                           */
    1  C/* SEGMENT NAME - SIREPT3                                    */
    1  C/*                                                           */
    1  C/* FUNCTION -  SIREPT3 WRITES TRIP AND VEHICLE DATA          */
    1  C/*                                                           */
    1  C/*                                                           */
    4     CONVERT DEBOARD/BOARD TIMES FROM CLOCK UNITS TO SECONDS

    4     WRITE DEBOARD/BOARD TIME SUMMARY

    4     VALIDATE TIMES

    4     WRITE DEBOARD/BOARD METHOD

    4     WRITE LAUNCH DELAY METHOD

    4     WRITE STATIC ENTRAINMENT OPTION

    4     WRITE SUMMARY OF VEHICLE ARRIVAL SOURCES

    1  C/*****************END OF SIREPT3*****************************/
```

```
PROJECT: AGT                    MEMBER: SIREPT4                 DATE
LIBRARY: USM                    LEVEL:  01.01                   TIME
TYPE:    PLI                    USERID: D425537                 PAGE
START
COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

1    C/*************************************************************************
1    C/*                                                                       *
1    C/* SEGMENT NAME -  SIREPT4                                               *
1    C/*                                                                       *
1    C/* FUNCTION - SIREPT4 WRITES THE SERVICE POLICY CHARACTERISTICS          *
1    C/*                                                                       *
1    C/*                                                                       *
4       CONVERT MERGE DELAY DISTRIBUTION TO CUMULATIVE PROBABILITY DIST.

4       WRITE MERGE DELAY DISTRIBUTION

4       CONVERT EMPTY VEH DELAY DIST. TO CUMULATIVE PROB. DIST.

4       WRITE EMPTY VEHICLE DELAY DISTRIBUTION

4       INCLUDE SIREPT4B TO WRITE EMPTY VEHICLE TECHNIQUE

4       WRITE-TRANSFER PROBABILITY

4       IF SERVICE IS SCHEDULED THEN

6         INCLUDE SIREPT4A TO WRITE VEHICLE SPACING AND HEADWAY

6         INCLUDE SIREPT4D TO WRITE ROUTE LISTS

4       ENDIF

4       WRITE SIMULATION CONTROL DATA (IN SIREPT4C)
6         SAMPLING INTERVAL
6         REPORT  INTERVAL
6         CHECKPOINT INTERVAL

1    C/*************************END OF SIREPT4*****************************/
```

```
PROJECT: ACT                    MEMBER: SIREPT4A                    DATE
LIBRARY: DSM                    LEVEL:  01.00                       TIME
TYPE:    PDL                    USERID: D425537                     PAGE
START
COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  C/*********************************************************************
   1  C/*                                                                   *
   1  C/* SEGMENT NAME - SIREPT4A                                           *
   1  C/*                                                                   *
   1  C/* FUNCTION - WRITE VEHICLE SPACING AND HEADWAY REPORT.              *
   1  C/*                                                                   *
   1  C/*                                                                   *
   1  C/*                                                                   *
   2       WRITE VEHICLE SPACING OPTION

   3       CONVERT HEADWAY TO SECONDS FROM CLOCK UNITS

   3       WRITE HEADWAY/ROUTE SUMMARY

   1  C/********************** END OF SIREPT4A*************************
```

```
PROJECT: AGT                    MEMBER: SIREPT4B                    DATE
LIBRARY: USM                    LEVEL:  01.00                       TIME
TYPE:    PDL                    USERID: 0425937                     PAGE
START
COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

1   C/*************************************************************
1   C/*                                                           *
1   C/* SEGMENT NAME - SIREPT4B                                   *
1   C/*                                                           *
1   C/* FUNCTION - SIREPT4B WRITES THE EMPTY VEHICLE DATA         *
1   C/*                                                           *
1   C/*                                                           *
4       WRITE EMPTY 'EMPTY VEHICLE NEEDED ELSEWHERE' PROBABILITY

4       WRITE EMPTY VEHICLE STORAGE POLICY

4       WRITE EMPTY VEHICLE RETRIEVAL POLICY INCLUDING
3           LIST OF STORAGE LINKS

4   C/*************************SIREPT4B*****************************
```

```
PROJECT: AGT                    MEMBER: SIREPT4D                    DATE
LIBRARY: DSM                    LEVEL:  01.00                       TIME
TYPE:    PLL                    USERID: D425537                     PAGE
START
COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

     C/*****************************************************************
     C/*                                                               *
     C/* SEGMENT NAME -  SIREPT4D                                      *
     C/*                                                               *
     C/* FUNCTION -  SIREPT4D WRITES THE ROUTE LIST SUMMARY            *
     C/*                                                               *
     C/*                                                               *
     C       WRITE A LIST OF ROUTE ASSIGNMENTS BY DESTINATION STATION

     C       WRITE TRIP-VEHICLE PROBABILITY DATA

     C       WRITE LIST OF STATIONS ON EACH ROUTE

     C/*                                                               *
     C/*****************SIREPT4D*******************************************
```

```
      PROJECT: ACT                    MEMBER: SIREPT5                    DATE
      LIBRARY: SSM                    LEVEL:  01.00                      TIME
      TYPE:    PLI                    USERID: D425537                    PAGE
STMT
 LVL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

   1  C/*****************************************************************
   1  C/*                                                               *
   1  C/* SEGMENT NAME - SIREPT5                                        *
   1  C/*                                                               *
   1  C/* FUNCTION - SIREPT5 WRITES THE STATION LINK SUMMARY            *
   1  C/*                                                               *
   1  C/*                                                               *
   4     DO FOR ALL STATION LINKS

   5       GET DEQUEUE OPTION IF MORE THAN 1 UPSTREAM LINK

   5       GET DIVERGE FUNCTION

   5       WRITE AND VALIDATE ALL DATA FOR THIS LINK(USE SIREPT5A-C)

   4     ENDDO
   1  C/*                                                               */
   1  C/*****************END OF SIREPT5*****************************/
```

```
      PROJECT: AGT                    MEMBER: SIREPT6                    DATE
      LIBRARY: DSM                    LEVEL:  01.00                      TIME
      TYPE:    PDL                    USERID: D426537                    PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1 C/*************************************************************** *
    1 C/*                                                               *
    1 C/* SEGMENT NAME - SIREPT6                                        *
    1 C/*                                                               *,
    1 C/* FUNCTION - SIREPT6 WRITES THE TRIP LINK SUMMARY               *,
    1 C/*                                                               *,
    1 C/*                                                               *,
    4     DO FOR ALL TRIP LINKS

    6       CONVERT LINK TIMES TO SECONDS FROM CLOCK UNITS

    6       WRITE AND VALIDATE ALL TRIP LINK DATA

    4     ENDDO
    1 C/*                                                               *,
    1 C/*************************END OF SIREPT6************************** /
```

```
      PROJECT: AGM                        MEMBER: SISCFG                        DATE
      LIBRARY: DSM                        LEVEL:  01.02                         TIME
      TYPE:    PLL                        USERID: B008654                       PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1 PROC: SISCFG  <STATION LINK CONFIGURATOR>
    2     <SISCFG BUILDS THE STRUCTURED DATA FOR THE DSM-MP FROM
    3     USER DEFINED STATION LINK PARAMETERS>
    4     <CONTROL IS PASSED TO THE CONFIGURATOR WHEN THE SLCFIG
    5     VARIABLE IS DEFINED IN THE SYSTEM CHARACTERISTICS DATA>
    6   FOR ALL LINKS
    7     DO
    8       ASSIGN LINK ID NUMBER AND QUANTITY PER LINK
    9       IF DOCK LINK
   10         SEPERATE BY DEBOARD/BOARD EVENT
   11       ENDIF
   12     ENDDO
   13   SET KNSL TO NUMBER OF LINKS
   14   ERROR CHECKS FOR CONFIGURATION
   15     ERROR IF NO STATION EXIT
   16     ERROR IF DOCK LINKS¬=
   17     ERROR IF QUEUE¬(1 TO M  OR  M  TO  M )
   18     ERROR IF QUEUQ OR DOCK > 10
   19   FOR EACH LINK
   20     DO  <BUILD STRUCTURED TABLES>
   21     LINK CAPACITY TO SLCAP
   22       IF LINK IS STORAGE
   23         SET LINK ID IN SLSTOR
   24       ENDIF
   25     LINK HEADWAYS TO SLHTA AND SLHTB
   26     LINK LINK-TYPE TO SLTYPE
   27     LINK DIVERGE TO SLDIVC
   28       IF TRAVEL TIME >0
   29         TRAVEL TIME / 60 #CSIZE TO SLTTIM
   30       ELSE
   31         IF LINK LENGTH > 0
   32           LL * SLVEL / 60 * CSIZETO SLTTIM
   33         ELSE
   34           0 TO SLTTIM
   35         ENDIF
   36       ENDIF
   37     LINK EVENTS TO SLEVL AND SET LINK POINTER FOR LINK
   38       ERROR IF EVENT ¬> EVENT + 1
   39       ERROR IF LAUNCH EVENT = 0
   40   ENDDO
   41   FOR EACH LINK
   42     DO <BUILD UPSTREAM CONNECTIVITY POINTERS AND LISTS>
   43       SET POINTER ENTRY FOR LINK IN SLUSP
   44       SET LINK ID'S FOR UPSTREAM KINKS IN SLUSL
   45       SET LIST RUN TIME SIZE IN KNSLU
   46     ENDDO
   47   FOR EACH LINK
   48     DO <BUILD DOWNSTREAM CONNECTIVETY POINTERS AND LISTS>
   49       SET POINTER ENTRY FOR LINK IN SLDSP
   50       SET LINK ID'S DOWNSTREAM LINKS IN SLDSL
   51       SET LIST RUN TIME SIZE IN KNSLD
   52     ENDDO
   53 END PROC
   54 <<NOTES:CONNECTIVITY DEFINITIONS
   55   --UPSTREAM LINKS          LINK TYPE          --DOWNSTREAM LINKS
   56     OL                        IR                   IS.MOB(IQ | DOCK)
   57     IS.SI.MIB | OL            IQ                   DOCK(D | D/B)
   58     IQ | IR.SI MIB | OL       DOCK (D)             DOCK (B)
   59     DOCK (D)                  DOCK (B)             DS.MOA(OQ | OR | DL)
   60     IQ | IR.SI.MIB | OL       DOCK (D/B | J)       DS.MOA(OQ | OR | DL)
   61     DOCK (B | D/B)            OQ                   OR | DL
   62     (OQ | DOCK).SO.MIA        OR                   DL
   63     DS.IS                     ST                   SI,SO
   64     IR                        IS                   ST
   65     ST                        SI                   IQ | DOCK
   66     DOCK (B | D/B)            DS                   ST
   67     SI                        SO                   OR
   68     SOURCE                    OL                   BL.(IR | IQ | DOCK)
   69     OL                        BL                   DL
   70     BL.(OR | OQ | DOCK)       DL                   SINK
```

```
      PROJECT: AOT                    MEMBER: SISCFG                    DATE
      LIBRARY: DSM                    LEVEL:  01.02                     TIME
      TYPE:    PDL                    USERID: B0C8654                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+--7

    5       SOURCE                  MIB                 IO | DOCK
    6       SOURCE                  MIA                 OR
    3       IN                      MOB                 SINK
    5       DOCK (S | D/R)          MOA                 SINK
    1   ENDROUTES>>
```

```
    PROJECT: AGT                    MEMBER: SITDGN
    LIBRARY: DSM                    LEVEL:  01.07
    TYPE:    PDL                    USERID: C120496
PART
COL  ----+----1----+----2----+----3----+----4----+----5----+---
    C/*************************************************************** */ 000100
    C/*                                                               */ 000200
  . C/* MODULE NAME - SITDGN                                          */ 000300
  1 C/*                                                               */ 000400
  1 C/* FUNCTION - DSM TRIP DEMAND GENERATION                         */ 000500
  1 C/*                                                               */ 000600
  1 C/* INCLUDED SEGMENTS -                                           */ 000700
  1 C/*       SITDGN2 - VERIFY AND INITIALIZE INPUT                   */ 000800
  1 C/*       SITDGN3 - GENERATE ERROR MESSAGES                       */ 000900
  1 C/*       SITDGN4 - WRITE TRIP GENERATION SUMMARY                 */ 001000
                                                                      001100
  1 PROC SITDGN: <DSM TRIP DEMAND GENERATION>                         001200
                                                                      001300
  4     INCLUDE SITDGN2 TO VERIFY INPUT                               001400
                                                                      001500
  4     IF                                                            001600
  6       EXPONENTIAL IAT OPTION  <DTIASW>                            001700
  4     THEN                                                          001800
                                                                      001900
  6       CALL SMRNG FOR RANDOM NUMBER                                002000
  6       ESTABLISH THE FIRST ARRIVAL TIME                            002100
 11             <DTARIV = -DTLMDA * LOG(DRANDM) +DTARIV>             002200
                                                                      002300
  4     ELSE  <USER INTERARRIVAL TIME DISTRIBUTION>                   002400
                                                                      002500
  6       CALL SMRSEL TO SAMPLE USER IAT DISTRIBUTION <DTIATD>        002600
  6       ESTABLISH THE FIRST ARRIVAL TIME                            002700
 11             <DTARIV = DTIATD(I,2) +DTARIV>                       002800
                                                                      002900
  4     ENDIF                                                         003000
                                                                      003100
  4     VERIFY THAT ARRIVAL<= END GENERATION TIME                     003200
  9         <DTARIV <= DTENDT>                                        003300
                                                                      003400
        INCLUDE SITDGN3 TO GENERATE ERROR MESSAGES                    003500
                                                                      003600
  .     WHILE                                                         003700
  6       ARRIVAL TIME <= END GEN  <DTARIV <= DTENDT>                 003800
  4     DO                                                            003900
                                                                      004000
  6       CALL SMRSEL TO SAMPLE  DESTINATION DIST.  <DTDESD>          004100
                                                                      004200
  6       CALL SMRSEL TO SAMPLE TRIP SIZE DIST.  <DTPASD>             004300
                                                                      004400
  6       WRITE A RECORD TO THE TRIP ARRIVAL FILE CONTAINING          004500
 11             ARRIVAL TIME                                         004600
 11             ORIGIN                                               004700
 11             DESTINATION                                          004800
 11             NUMBER OF PASSENGERS                                 004900
                                                                      005000
  6       SAVE STATISTICS                                             005100
                                                                      005200
  6       IF                                                          005300
  8         EXPONENTIAL ARRIVAL RATE  <DTIASW>                        005400
  6       THEN                                                        005500
                                                                      005600
  8         CALL SMRNG FOR A RANDOM NUMBER                            005700
  8         ESTABLISH TIME OF NEXT ARRIVAL                            005800
 13               <DTARIV = DTARIV - DTLMDA * LOG(DRANDM)>             005900
                                                                      006000
  6       ELSE  <USER INTERARRIVAL TIME DISTRIBUTION>                 006100
                                                                      006200
  8         CALL SMRSEL TO SAMPLE USER IAT DIST. <DTIATD>              006300
  8         ESTABLISH TIME OF NEXT ARRIVAL                            006400
 13               <DTARIV = DTARIV + DTIATD(I,2)>                      006500
                                                                      006600
  6       ENDIF                                                       006700
                                                                      006800
  .     ENDDO                                                         006900
                                                                      007000
```

```
   PROJECT: AGT                    MEMBER: SITDGN
   LIBRARY: DSM                    LEVEL:  01.07
   TYPE:    PDL                    USERID: C120496
ART
OL ----+----1----+----2----+----3----+----4----+----5----
      INCLUDE SITDGN4 TO WRITE A TRIP GENERATION REPORT CONTAINING          007100
           INPUT DATA AND                                                   007200
>          GENERATION SUMMARY                                               007300
                                                                            007400
3    END?ROC SITDGN                                                         007500
1  C/**********************END OF SITDGN*****************************/      007600
```

```
      PROJECT: ACT                    MEMBER: SITDGN2                  DATE:
      LIBRARY: DSM                    LEVEL:  01.03                    TIME:
      TYPE:    PDL                    USERID: D425537                  PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  C/*****************************************************************/
    1  C/*                                                               */
    1  C/* SEGMENT NAME - SITDGN2                                        */
    1  C/*                                                               */
    1  C/* FUNCTION - VERIFY AND INITIALIZE INPUT FOR DSM TRIP GEN       */
    1  C/*                                                               */
    1  C/*                                                               */

    4      FOR
    6        FOR DEST. 1 THRU KNS
    4      DO

    6        INITIALIZE STATISTICS COUNTERS FOR
   11              NUMBER OF PASSENGERS AND
   11              NUMBER OF TRIPS
    4      ENDDO

    4      FOR
    6        ALL ERROR TYPES
    4      DO
    6        INITIALIZE ERROR INDICATOR
    4      ENDDO

    4      CALL SICOMP TO CHANGE DESTINATION DIST.  <DTDESD>
    9           TO CUMULATIVE PROB. DIST.

    4      IF
    6        USER IAT OPTION  <DTIASW>
    4      THEN

    6        GET MEAN OF USER IAT DIST    <DTIATD>

    6        CALL SICOMP TO CHANGE IAT DIST. TO
   11              CUMULATIVE PROB. DIST.

    4      ELSE    <EXPONENTIAL DISTRIBUTION>

    6        CHECK REASONABLENESS OF MEAN IAT  <DTLMDA>

    4      ENDIF

    4      CALL SICOMP TO CHANGE TRIP SIZE DIST.    <DTPASD>
    9           TO CUMULATIVE PROB. DIST.

    4      CHECK REASONABLENESS OF END GEN. TIME  <DTENDT>

    4      VERIFY SIMULATED STATION ID  <DTSTAN>

    1  C/*************************END OF SITDGN2*************************/
```

```
      PROJECT: AOT                        MEMBER: SITDGN3                        DATE:
      LIBRARY: DSM                        LEVEL:  01.03                          TIME:
      TYPE:    PL1                        USERID: D425537                        PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  C/*****************************************************************/
    1  C/*                                                               */
    1  C/* SEGMENT NAME - SITDGN3                                        */
    1  C/*                                                               */
    1  C/* FUNCTION - GENERATE ERROR MESSAGES FOR DSM TRIP GEN           */
    1  C/*                                                               */

    4      FOR
    6        ALL ERROR TYPES
    4      DO

    6        IF
    6          ERROR FLAG IS ON
    6        THEN

    8          CALL ERROR TO WRITE ERROR MESSAGE

    6        ENDIF

    4      ENDDO

    4      IF
    6        ANY SERIOUS ERROR
    4      THEN

    6        CALL ERROR TO WRITE TERMINATION MESSAGE AND
   11             END RUN

    4      ENDIF

    1  C/*****************************END OF SITDGN3*************************/
```

```
      PROJECT: AGT                          MEMBER: SITDGN4                          DATE
      LIBRARY: DSM                          LEVEL:  01.02                            TIME
      TYPE:    PDL                          USERID: 8435537                          PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-
    1 C/*************************************************************************/
    1 C/*                                                                       */
    1 C/* SEGMENT NAME - SITDGN4                                                */
    1 C/*                                                                       */
    1 C/* FUNCTION - WRITE DSM TRIP GENERATION SUMMARY                          */
    1 C/*                                                                       */
    9          WRITE SUMMARY  OF INPUT AND ACTUAL
   14               MEAN ARRIVAL TIME AND TRIP SIZE

    9          STATION COUNT = 1
    9          WHILE
   11            STATION COUNT < MAX STATIONS  <KNS>
    9          DO

   11             WRITE HEADER
   11             WHILE
   13               STATION COUNT < MAX STATIONS  <KNS>   AND
   13               LINECOUNT < 50
   11             DO

   13               WRITE 1 LINE SUMMARY OF TRIPS DESTINATING AT A GIVEN
   15                    STATION

   11             ENDDO

    9          ENDDO

    9          WRITE TOTAL TRIPS GENERATED

    1 C/*************************************END OF SITDGN4*********************/
```

```
PROJECT: ACT                    MEMBER: SIVDGN                    DATE:
LIBRARY: DSM                    LEVEL:  01.06                     TIME:
TYPE:    PDL                    USERID: 0425537                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--

   1  C/*************************************************************************/
   1  C/*                                                                       */
   1  C/* MODULE NAME - SIVEGN                                                  */
   1  C/*                                                                       */
   1  C/* FUNCTION - GENERATE DSM VEHICLE ARRIVAL FILE                          */
   1  C/*                                                                       */
   1  C/* INCLUDED SEGMENTS -                                                   */
   1  C/*       SIVDGN2 - VERIFY AND INITIALIZE INPUT VALUES                    */
   1  C/*       SIVDGN4 - GENERATE SCHEDULED VEHICLE                            */
   1  C/*       SIVDGN5 - GENERATE DEMAND RESPONSIVE VEHICLE                    */
   1  C/*       SIVDGN6 - GENERATE ONBOARD TRIPS                                */
   1  C/*       SIVDGN7 - WRITE VEHICLE GENERATION REPORT                       */
   1  C/*                                                                       */
   2   PROC SIVDGN:  <DSM VEHICLE DEMAND GENERATION>

   4      INCLUDE SIVDGN2 TO VERIFY INPUT


   4      WHILE
   9        NEXT ARRIVAL <= END TIME
  11             <DVARIV <= DVENDT>
   4      DO


   6        IF
   6          SERVICE POLICY IS SCHEDULED  <DVSERV=3>
   7         THEN

   8          INCLUDE SIVDGN4 TO GENERATE SCHEDULED VEHICLE

   6        ELSE  <DEMAND RESPONSIVE>

   8          INCLUDE SIVDGN5 TO GENERATE DEMAND RESPONSIVE VEHICLE

   6        ENDIF

   6        CALL SMRSEL TO SAMPLE SINK DISTRIBUTION <DVSNKD>


   6        FOR
   5          WHOLE TRAIN  <I TO DVTRLN>
   6        DO

   5          IF
  10            TRAIN IS STOPPING   <DVNXST = DVSTAN>
   5          THEN

  10            INCLUDE SIVDGN6 TO GENERATE ONBOARD TRIPS

  10            WRITE TO THE VEHICLE ARRIVAL FILE A RECORD CONTAINING
  15                  ARRIVAL TIME
  15                  NEXT STOP
  15                  DIVERT TO STORAGE INDICATOR
  15                  SINK
  15                  ROUTE
  15                  NUMBER OF PASSENGERS  ONBOARD
  15                  TRAIN LENGTH
  15                  NUMBER OF TRIPS ONBOARD

  10            IF
  12              THIS IS 1ST VEH IN TRAIN   <IVEH = 1>
  10            THEN

  12              IF
  14                TRAIN IS STOPPING   <DVNXST = DVSTAN>
  12              THEN
  14                COLLECT STATISTICS ON STOPPING VEHICLES AND TRIPS
  12              ELSE
  14                COLLECT STATISTICS ON VEHICLES NOT STOPPING
  12              ENDIF
```

```
      PROJECT: AGT                          MEMBER: SIVDGN                          DATE
      LIBRARY: DSM                          LEVEL:  01.08                           TIME
      TYPE:    PLL                          USERID: 0425537                         PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  10            ENDIF

  10            IF
  12              VEH IS STOPPING AND HAS ONBOARD TRIPS
  17                    <DVNXST = DVSTAN AND DVVTRP > 0>
  10            THEN

  12              WRITE TO THE VEHICLE ARRIVAL FILE ALL THE
  17                    TRIP FOLLOWER RECORDS CONTAINING
  19                      ORIGIN
  14                      DESTINATION
  16                      NUMBER OF PASSENGERS

  10            ENDIF

   8          ENDIF

   6        ENDDO

   4      ENDDO

   4      INCLUDE SIVDGN7 TO WRITE VEHICLE GENERATION REPORT CONTAINING
   9           INPUT DEMAND SUMMARY
   9           VEHICLE GENERATION SUMMARY

   2   ENDPROC SIVDGN
   1  C/**************************END OF SIVDGN*******************************/
```

```
PROJECT: AGT                    MEMBER: SIVDGN2                    DATE
LIBRARY: DSM                    LEVEL:  01.00                      TIME
TYPE:    PLL                    USERID: D425537                    PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

   1  C/*************************************************************************
   1  C/*                                                                       *
   1  C/* SEGMENT NAME - SIVDGN2                                                *
   1  C/*                                                                       *
   1  C/* FUNCTION - DSM VEHICLE GEN. INPUT VERIFICATION                        *
   1  C/*                                                                       *

   4      FOR
   6        ALL ERRORS
   4      DO
   6        INITIALIZE AN ERROR INDICATOR
   4      ENDDO

   4      IF
   6        SCHEDULED SERVICE  <DVSERV=3>
   4      THEN

   6        FOR
   8          ROUTES 1 TO MAX  <KNR>
   6        DO
   8          INITIALIZE STATISTICS ARRAY FOR
  10               TRIP AND VEHICLE INFORMATION
   6        ENDDO

   4      ENDIF

   4      CHECK REASONABLENESS OF END GENERATION TIME  <DVENDT>

   4      IF
   6        SYNCHRONOUS ENVIRONMENT   <DVSNSW = 1>
   4      THEN
   6        VALIDATE SLOT LENGTH    <DVSLOT>
   4      ENDIF

   4      IF
   6        SERVICE IS SCHEDULED  <DVSERV = 3>
   4      THEN

   6        ROUTE = 1
   6        WHILE
   8          ROUTE <= MAX ROUTES <KNR> AND
   8          NO ERRORS FOUND
   6        DO
   8          VALIDATE TRAIN LENGTH FOR ROUTE   <DVTLNR>
   6        ENDDO

   6        IF
   8          NEXT STOP CHOSEN FROM ROUTE LIST <DVNXID=1>
   6        THEN

   8          ROUTE = 1
   8          WHILE
  11            ROUTE <= MAX ROUTES <KNR> AND
  11            NO ERRORS FOUND
   8          DO
  10            VERIFY THAT EACH ROUTE ENTRY <DVRSCH>
   8          ENDDO

   6        ENDIF

   4      ENDIF


   4      IF
   6        SERVICE IS SCHEDULED  <DVSERV=3>
   4      THEN

   6        FOR
   8          ROUTE = 1 TO MAX ROUTES <KNR>
   6        DO
```

```
      PROJECT: AGT                    MEMBER: SIVDGN2                  DATE
      LIBRARY: USM                    LEVEL:  01.06                    TIME
      TYPE:    PAL                    USERID: 0425537                  PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

*
         CALL SIGIAT TO DETERMINE 1ST ARRIVAL <DVARIV>
         SAVE FIRST ARRIVAL FOR ROUTE  <DVARVR(ROUTE)>
         SAVE EARLIEST ARRIVAL TIME OF ALL ROUTES
       ENDDO

     ELSE  <DEMAND RESPONSIVE>

       CALL SIGIAT TO DETERMINE FIRST ARRIVAL TIME FOR
            DEMAND RESPONSIVE SERVICE <DVARIV>

     ENDIF


     VERIFY THAT FIRST ARRIVAL TIME DOES NOT EXCEED END TIME
          <DVARIV <= DVENDT>

     INCLUDE SIVDGN2B TO WRITE ERROR MESSAGES

C/*********************END OF SIVDGN2*****************************
```

```
      PROJECT: AGT                    MEMBER: SIVDGN2A                    DATE
      LIBRARY: DSM                    LEVEL:  01.00                       TIME
      TYPE:    PDL                    USERID: D425537                     PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1 C/*************************************************************************
    1 C/*                                                                       *
    1 C/* SEGMENT NAME - SIVDGN2A                                               *
    1 C/*                                                                       *
    1 C/* FUNCTION - CONVERT PROBABILITY DISTRIBUTIONS TO CUMULATIVE            *
    1 C/*              PROBABILITY DISTRIBUTIONS                                *


    4      IF
    6        USER IAT DIST. USED  <DVIASW=I>
    5       THEN

    6        COMPUTE MEAN IAT
    6        CALL SICUMP TO CHANGE USER IAT DIST.  <DVIATD>
   10             TO CUMULATIVE PROB. DIST.

    4      ELSE <EXPONENTIAL ARRIVAL RATE>

    6        CHECK MEAN IAT'S    <DVLMDA>

    4      ENDIF


    4      CALL SICUMP TO CHANGE DESTINATION DIST. TO CUMULATIVE PROB. DIST.

    4      CALL SICUMP TO CHANGE TRIP SIZE PROBABILITY DIST. TO CUMULATIVE
    9           PROB. DIST.

    4      IF
    6        SERVICE IS DEMAND RESP.  <DVSERV = 1 OR 2>
    4      THEN

    6        CALL SICUMP TO CHANGE TRAIN LENGTH PROBABILITY
   11             DIST. TO CUMULATIVE PROB. DIST.<DVTLND>

    4      ENDIF


    4      CALL SICUMP  TO CHANGE TRIPS/VEHICLE DIST. TO CUMULATIVE
    9           PROB. DIST.<DVTRPD>

    4      CALL SICUMP TO CHANGE SINK DIST. TO CUMULATIVE PROB. DIST.
   10            <DVSNKD>

    1 C/*************************END OF SIVDGN2*****************************/
```

```
      PROJECT: AGT                    MEMBER: SIVDGN2B                  DATE
      LIBRARY: LSM                    LEVEL:  01.01                     TIME
      TYPE:    PL1                    USERID: D425537                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1 C/*************************************************************************
    1 C/*                                                                       *
    1 C/* SEGMENT NAME - SIVDGN2B                                               *
    1 C/*                                                                       *
    1 C/* FUNCTION - SIVDGN2B IS INCLUDED BY DSM VEH DEMAND GENERATION TO       *
    1 C/*              GENERATE ANY ERROR MESSAGES FOUND BY VEH GENERATION.     *

    4     FOR
    6       ALL ERROR TYPES
    4     DO

    6       IF
    6         ERROR FOUND
    6       THEN

    6         CALL ERROR TO WRITE ERROR MESSAGE

    6       ENDIF

    4     ENDDO

    4     IF
    6       SERIOUS ERROR FOUND
    4     THEN

    6       CALL ERROR TO WRITE TERMINATION MESSAGE AND
   11           END RUN

    4     ENDIF
    1 C/*****************************END OF SIVDGN2B*****************************/
```

```
PROJECT: AGT                    MEMBER: SIVDGN4                    DATE
LIBRARY: GSM                    LEVEL:  01.03                      TIME
TYPE:    PLI                    USERID: D425537                    PAGE
START
COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

   1 C/*************************************************************/
   1 C/*                                                           */
   1 C/* SEGMENT NAME - SIVDGN4                                    */
   1 C/*                                                           */
   1 C/* FUNCTION - GENERATE VEHICLE FOR SCHEDULED SERVICE POLICY  */
   1 C/*                                                           */

   5          ARRIVAL TIME AND ROUTE HAVE ALREADY BEEN DETERMINED
  10               <DVARIV,DVROUT>

   5          DETERMINE TRAIN LENGTH FOR ROUTE   <DVTLNR(ROUTE)>

   8          IF
  10            NEXT STOP CHOSEN BY SWITCH  <DVNXID=1>
   6          THEN

  10            IF
  11 -            SWITCH IS ON   <DVRTST>
  10            THEN
  12              THE NEXT STOP IS THE SIMULATED STATION
  16                   <DVNXST=DVSTAN>
  10            ENDIF

   8          ELSE    <SCAN ROUTE LIST>

  10            SCAN ROUTE LIST   <DVRSCH>

  10            IF
  12              SIMULATED STATION IS ON THE ROUTE
  10            THEN
  12              NEXT STOP IS SIMULATED STATION
  17                   <DVNXST = DVSTAN>
  10            ENDIF

   8          ENDIF

   6          CALL SIGIAT TO DETERMINE THE NEXT INTERARRIVAL TIME
  15               FOR THE SELECTED ROUTE  <DVARVR(ROUTE)>
   -          SCAN THE LIST OF NEXT ARRIVAL TIME BY ROUTE TO FIND
  15               THE TIME OF THE NEXT ARRIVAL  <DVARIV>

   1 C/*****************************END OF SIVDGN4*********************/
```

```
      PROJECT: AGT                    MEMBER: SIVDGNS                    DATE:
      LIBRARY: DSM                    LEVEL:  01.02                      TIME:
      TYPE:    PDL                    USERID: D425537                    PAGE:
START
COL   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1 C/*************************************************************************/
    1 C/*                                                                       */
    1 C/* SEGMENT NAME - SIVDGNS                                                */
    1 C/*                                                                       */
    1 C/* FUNCTION - GENERATE VEHICLE FOR DEMAND RESPONSIVE SERVICE POLICY      */
    1 C/*                                                                       */
    5         CALL SMRSEL TO SAMPLE TRAIN LENGTH DIST.  <DVTLND>

    5         CALL SMRNG FOR RANDOM NUMBER     <DRANDO>
    6         TRAIN WILL STOP IF RANDOM NO. IS >= P(NEXT STOP =SIM*D STATION)
   13             <DRANDO >= DVSTPP>

    5         CALL SIGIAT FOR NEXT IAT  <DVIATM>
    1 C/*************************END OF SIVDGNS*********************************/
```

```
PROJECT: AGT                    MEMBER: SIVDGN6                    DATE:
LIBRARY: DSM                    LEVEL:  01.04                      TIME:
TYPE:    PLI                    USERID: D425537                    PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1   C/*****************************************************************/
    1   C/*                                                               */
    1   C/* SEGMENT NAME - SIVDGN6                                        */
    1   C/*                                                               */
    1   C/* FUNCTION - DSM GENERATION OF ONBOARD TRIPS                    */
    1   C/*                                                               */

   10            CALL SMRSEL TO SAMPLE TRIPS/VEHICLE DIST.  <DVTRPD>

   10            DVVTRP = NO. TRIPS CHOSEN
   10            WHILE
   12              TOTAL TRIPS <= NO. TRIPS CHOSEN AND
   12              PASSENGERS <= VEH CAPACITY
   17                   <TOTAL TRIPS <= DVVTRP AND DVVPAS <= DVCAPP>
   10            DO

   12              CALL SMRSEL TO SAMPLE DESTINATION DIST.  <DVDESD>
   12              CALL SMRSEL TO SAMPLE TRIP SIZE DIST.  <DVPASD>
   12              TOTAL TRIPS = TOTAL TRIPS + 1
   12              PASSENGERS = PASSENGERS + TRIP SIZE
   17                   <DVVPAS = DVVPAS + DVTRIP>

   10            ENDDO

    1   C/*********************END OF SIVDGN6*****************************/
```

```
      PROJECT: AGT                        MEMBER: SIVDGN7                        DATE:
      LIBRARY: DSM                        LEVEL:  01.01                          TIME:
      TYPE:    PLL                        USERID: D425537                        PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-
    1  C/*************************************************************************/
    1  C/*                                                                       */
    1  C/* SEGMENT NAME - SIVDGN7                                                */
    1  C/*                                                                       */
    1  C/* FUNCTION - WRITE VEHICLE GENERATION SUMMARY REPORT                    */
    1  C/*                                                                       */
    9          IF
   11            SERVICE IS SCHEDULED   <DvSERV=3>
    9          THEN

   11            ROUTE = 1
   11            WHILE
   13              ROUTE <= MAX ROUTES <KNR>
   11            DO

   13              WRITE HEADER
   13              WHILE
   15        -       ROUTE <= MAX ROUTES  <KNR> AND
   15                LINECOUNT <= 50
   13              DO

   15                WRITE 1 LINE SUMMARIZING INPUT AND ACTUAL VALUES FOR
   20                      THE ROUTE
   13              ENDDO

   11            ENDDO

   11            ROUTE = 1
   11            WHILE
   13              ROUTE <= MAX ROUTES <KNR>
   11            DO

   13              WHILE
   15                ROUTE <= MAX ROUTES <KNR>
   15                LINECOUNT <= 50
   13              DO

   15                WRITE 1 LINE SUMMARY OF VEHICLES GENERATED FOR THE ROUTE
   13              ENDDO

   11            ENDDO

   11            WRITE TOTAL VEHICLES FOR ALL ROUTES

    9          ELSE          <DEMAND RESPONSIVE SERVICE>

   11            WRITE SUMMARY OF INPUT AND ACTUAL VALUES USED

   11            WRITE SUMMARY OF VEHICLES GENERATED

   11            WRITE TOTAL VEHICLES GENERATED

    9          ENDIF
    1  C/*************************END OF SIVDGN7*******************************/
```

```
      PROJECT: AGT                    MEMBER: SLIST                        DATE
      LIBRARY: DSM                    LEVEL:  00.01                        TIME
      TYPE:    PDL                    USERID: H639547                      PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC LIST <LIST SAMPLED ITEMS OR PRODUCE STATISTICAL SUMMARY>
    3    <LIST PRINTS OUT THE CONTENTS OF ANY SPECIFIED BIN, LISTING
    4     EVERY KTH ELEMENT. THE BIN TO BE PRINTED IS NBIN(1). K IS GIVEN BY
    4     PAR(1).  IF EVERY ELEMENT IS TO BE LISTED, PAR(1) MAY BE LEFT BLANK
    4     AND IT IS ASSUMED THAT PAR(1) = 1>
    4     I1 = LOWER INDEX OF BIN
    4     I2 = HIGHER INDEX OF BIN
    4     IF
    6       PAR(1)>1
    4     THEN <STATISTICAL SUMMARY DATA>
    6       COMPUTE SUM, MEAN & STANDARD DEVIATION OF DATA
    6       DISPLAY SUMMARY
    4     ELSE <PRODUCE A LIST OF SAMPLED ITEMS>
    6       FOR
    8         I=I1 TO I2
    6       DO
    9          DISPLAY VALUE FROM BIN AREA
    6       ENDDO
    4     ENDIF
  - 1  ENDPROC
```

```
      PROJECT: AOT                          MEMBER: SMACRO                        DATE:
      LIBRARY: OSM                          LEVEL:  01.02                         TIME:
      TYPE:    PDL                          USERID: P326507                       PAGE:
START
 COL  ---+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROVIDES A COMMON MEMBER THAT BRINGS INTO COMPILATION THE
    3    FOLLOWING SIMULATION MACROS:
    9           DBUG
    9           DCUE
    9           DCUEM
    9           DCUEMID
    9           ENDQLOOP
    9           FREE
    9           GET
    9           MULTICK
    9           NQUE
    9           QLOOP
    9           SCHED
    9           SSPMAC
    9           SUPMAC
    9           VRAND
    9           VRANDN
```

```
      PROJECT: ACT                    MEMBER: SMBRD                    DATE:
      LIBRARY: SSM                    LEVEL:  01.19                    TIME:
      TYPE:    PAL                    USERID: P326507                  PAGE:
START
 COL  ----+----.----+----2----+----3----+----4----+----5----+----6----+----7-

    1  LOCAL VAR NAME  DIM    DESCRIPTION
    1  --------------- ------ ------------------------------------------------
    1  MATCH           -      INDICATES WHETHER OR NOT THE VEH & TRIP ARE
   20                          COMPATIBLE
   20                          1===>MATCH
   26                          2===>NO MATCH
    1  HEADT           -      FIRST TRIP IN BOARDING QUEUE
    1  TID             -      CURRENT TRIP BEING PROCESSED
    1  HEADV           -      FIRST VEHICLE IN THE TRAIN
    1  VID             -      CURRENT VEHICLE BEING PROCESSED
    1  CHECK           -      INDICATES WHETHER OR NOT THE COMPATIBILITY OF
   20                          TRIP AND VEHICLE HAS BEEN TESTED
   20                          T===>WAS TESTED THEREFORE PROCEED TO NEXT TRIP
   20                          F===>WAS NOT TESTED, THEREFORE TRY THE NEXT
   31                              VEHICLE
    1  ------------------------------------------------------------------------
    1  PROC:SMBRD(V)
    3    VTOTP(V):=0
    3    CASEENTRY(PULSER)

    5    CASE(DEMAND RESPONSIVE SINGLE PARTY)
    5      IF   <THERE IS A TRIP ON THE VEHICLE AND THE VEHICLE IS IN THE
    7        BOARD EVENT, THERE IS AN ERROR>
    7        VTRIPQ(V)¬=0  AND VMEVNT(V) = 4
    3      THEN
    7        ERROR--TERMINATE
    5      ELSE
    7        IF <THERE IS TRIP IN BOARDING QUEUE>
    9          SBQTL¬=0
    7        THEN
    9          T:=TQUECH(SBQTL)       <SET T TO TRIP AT HEAD OF BOARDING  QUEUE.
    9          DQ T FROM SBQTL        <REMOVE T FROM BOARDING QUEUE>
    9          NQ T INTO VBLTL(V) <INSERT T INTO BOARDING LIST OF V>
    9          VTOTP(V):=TPASS(T)
    7        ENDIF
    5      ENDIF

    5    CASE(DEMAND RESPONSIVE MULTIPARTY)
    5      IF   <THERE IS A TRIP IN THE BOARDING QUEUE>
    7        SBQTL¬=0
    5      THEN
    7        IF   <THE VEHICLE IS EMPTY>
    6          VNPASS (V) = 0
    7        THEN
    9          T:=TQUECH(SBQTL)       <SET T TO TRIP AT HEAD OF BOARDING  QUEUE>
    9          TID:= TQUECH (T)
    9          DQ T FROM SBQTL        <REMOVE T FROM BOARDING QUEUE>
    9          NQ T INTO VBLTL(V) <INSERT T INTO BOARDING LIST OF V>
    9          VTOTP(V):=TPASS(T)     <SET THE # OF BOARDING PASS TO SIZE OF T>
    7        ENDIF
```

```
PROJECT: AGT                    MEMBER: SMORD                    DATE:
LIBRARY: DSM                    LEVEL:  01.19                    TIME:
TYPE:    PDL                    USERID: P32600?                  PAGE:
START
 COL  ---+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   7        WHILE  THERE ARE MORE TRIPS IN THE BOARDING QUEUE AND MORE
   9          SPACE ON THE VEHICLE
   9          SEQTL ¬= 0
   9          VNPASS(V) + VTOTP (V) < VCAP
   7        FOR
   9          EACH TRIP TID IN BOARDING QUEUE
   7        DO
   9          IF   <TID CAN FIT ON VEHICLE>
  11            VCAP >= VNPASS(V) + VTOTP (V)+TPASS(TID)
   9          THEN
  11            MATCH:=SMRSEL(PCOMPD) <DETERMINE COMPATIBILITY>
  11            IF   <COMPATIBLE>
  13              MATCH=1
  11            THEN
  13              DQ TID FROM SEQTL <REMOVE T FROM BOARDING QUEUE>
  13              NQ TID INTO VBLTL(V) <INSERT I INTO BOARDING LIST OF V>
  13              VTOTP(V):=VTOTP(V)+TPASS(TID) <INC # OF BOARDIN PASS.>
  11            ENDIF
   9          ENDIF
   7        ENDDO
   5      ENDIF

   3    CASE(SCHEDULED SERVICE)
   5      IF   <THERE ARE TRIPS WAITING IN BOARDING QUEUE>
   7        SEQTL¬=0
   5      THEN
   7        HEADT = TQUECH (SEQTL)
   7        TID =^HEADT
   7        DO
   9          HEADV = V
   9          VID = V
   9          DO
  11            IF
  13              VCAP >= VNPASS (VID) + VTOTP (VID) + TPASS (TID)
  11            THEN
  13              CHECK:= TRUE
  13              CASEENTRY(COMPATIBILITY METHOD)
  13              CASE(PROBABILITY OF COMPATIBILITY)
  15                MATCH:=SMRSEL(PCOMPD)
  13              CASE(TABLE OF STATIONS FOR EACH ROUTE)
  15                IF
  17                  THE ROUTE ASSIGNMENT TABLE IS USED
  15                THEN
  17                  IF
  19                    THE ROUTE OF THE VEHICLE MATCHES THE ROUTE
  19                    ASSIGNED TO THE TRIP'S DESTINATION
  17                  THEN
  19                    MATCH = 1
  17                  ELSE
  19                    MATCH = 2
```

```
      PROJECT: ACT                    MEMBER: SMBRD                    DATE
      LIBRARY: LSM                    LEVEL:  01.19                    TIME
      TYPE:    PDL                    USERID: P326507                  PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  17                  ENDIF
  15                ELSE
  17                  IF
  19                    DEST OF T IS ON THE ROUTE OF V
  17                  THEN
  19                    MATCH:=1
  17                  ELSE
  19                    MATCH:=2
  17                  ENDIF
  15                ENDIF
  13              ENDCASE
  15              IF  <COMPATIBLE>
  15                MATCH=1
  13              THEN
  15                IF
  17                  TID =HEADT
  15                THEN
  17                  HEADT = TQUEUE (TID)
  15                ENDIF
  15                DO TID FROM SEQTL <REMOVE T FROM BOARDING QUEUE>
  15                DO TID INTO VEHTL(V) <INSERT T INTO BOARDING LIST OF V>
  15                VTOTP(V):=VTOTP(V)+TPASS(TID) <INC # OF BOARDIN PASS.>
  13              ENDIF
  11            ELSE
  13              CHECK:=FALSE
  11            ENDIF
  11            VID = VTRNCH (VID)
   9          UNTIL
  11            VID = HEADV OR <LAST VEHICLE OF A TRAIN IS CHECKED>
  11            VID =0 <THE ONLY VEHICLE IS CHECKED> OR
  11            CHECK=TRUE <START OVER WITH THE NEXT TRIP AND FIRST VEHICLE>
   9          ENDDO
   9          IF
  11            SEQTL ¬= 0
   9          THEN
  11            TID = TQUEUE (TID)
   9          ENDIF
   7        UNTIL
   9          SEQTL = 0 OR
   9          TID = HEADT
   7        ENDDO
   5      ENDIF
   3    ENDCASE
   1  ENDPROC
```

```
      PROJECT: AGT                          MEMBER: SMOBRD                          DATE
      LIBRARY: DSM                          LEVEL:  01.08                           TIME
      TYPE:    PEL                          USERID: P326507                         PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  LOCAL VAR NAME  DIM     DESCRIPTION
    1  --------------- ------ ----------------------------------------------------
    1  XFER            -       INDICATES WHETHER OR NOT TRIP WILL TRANSFER
   20                           1===>TRANSFER
   20                           2===>NO TRANSFER
    1  ----------------------------------------------------------------------------
    1  PROC:SMOBRD(V)
    3    <FROM THE TRIPS ON BOARD V, BUILD 2 LISTS:
    5      1. THOSE TRIPS THAT DEBOARD AND LEAVE THE SYSTEM; AND
    5      2. THOSE TRIPS THAT DEBOARD TO TRANSFER
    4     AND COUNT TOTAL NUMBER OF PASSENGERS DEBOARDING>
    3    VTOTP(V):=0
    2    FOR
    5      EACH TRIP(T) ON VEHICLE V
    3    DO
    5      IF
    7        TDEST(T)=STSIM
    8      THEN
    7        DO T FROM VTRIPQ(V)
    7        NQ T INTO VDSLTL(V)
    7        VTOTP(V):=VTOTP(V)+TPASS(T)
    8      ELSE
    7        IF   <TRANSFER OPTION IN USE>
    9          PXFERI=T
    7        THEN
    9          RUN SMRSEL (PXFERD,XFER)
    9          IF
   11            XFER=1
    9          THEN
   11            DO T FROM VTRIPQ(V)
   11            NQ T INTO VDXLTL(V)
   11            VTOTP(V):=VTOTP(V)+TPASS(T)
    9          ENDIF
    7        ENDIF
    5      ENDIF
    3    ENDDO
    1  ENDPROC
```

```
    PROJECT: AGI                    MEMBER: SMDETR                    DATE:
    LIBRARY: USM                    LEVEL:  01.02                     TIME:
    TYPE:    PLC                    USERID: P316507                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  ENGL: SMDETR(V) <DETRAIN ALL VEHICLES BEHIND THIS LEAD VEHICLE IN THE
   2    TRAIN AND ADD THEM TO THE STATION LINK MEMBERSHIP LIST>
   3    SET TEMPORARY MEMBERSHIP LIST TAIL EQUAL TO THE SLMEMT(VSL(V))
   4    IF V IS THE ACTIVE VEHICLE (CAME OFF THE FEL), THEN LABEL ALL VEHICLE
   5    IN THE TRAIN USING VBEVCH(VEHICLE)>
   6    IF
   5      V = VACTIV
   6    THEN
   6      FOR
   7        EACH VEHICLE IN THE TRAIN OF V
   6      WHILE
   7        NOT ON THE LEAD VEHICLE
   8      DO
   7        LABEL ALL VEHICLES ON THIS TRAIN AS HAVING COME OFF THE FEL
   7        VBEVCH (VID):= KMX + 1
   5      ENDDO
   3    ENDIF
   3    FOR
   5      EACH VEHICLE ON THE MEMBERSHIP LIST
   6    DO
   5      DEQUE IT FROM THE STATION LINK MEMBERSHIP LIST
   5      NQUE IT ONTO THE TEMPORARY MEMBERSHIP LIST
   3    UNTIL
   5      THE LEAD VEHICLE OF THE TRAIN HAS BEEN TRANSFERED
   3    ENDDO
   3    FIND THE TAIL OF THE TRAIN
   3    SET A TEMPORARY TAIL EQUAL TO THE TAIL
   3    WHILE
   5      VEHICLES REMAIN ON THE TRAIN
   3    FOR
   5      EACH VEHICLE BEGINNING WITH THE LEAD ONE
   3    DO
   5      DEQUE THE VEHICLE FROM THE TRAIN CHAIN
   6      IF
   7        THE VEHICLE JUST DEQUEUED IS NOT THE LEAD VEHICLE IN
   7        THE TRAIN
   5      THEN
   7        ENQUE THE VEHICLE INTO THE TEMPORARY MEMBERSHIP LIST
   7        LIST IT ON ADLST
   7        REFRESH THE FOLLOWING VARIABLES IN THE IMAGE OF V:
   9          VTIME, VMEVNT, VSL, VCURR, VNXSTN, VDIVST, VSINK, VGREAS.
   5      ENDIF
   5      VTRLEN(V):=1
   3    ENDDO
   3    END THE LIST OF DETRAINED VEHICLES WITH A ZERO
   3    WHILE
   5      VEHICLES REMAIN ON THE STATION LINK
   3    DO
   5      DEQUE VEHICLES FROM THE STATION LINK MEMBERSHIP LIST
   5      NQUE VEHICLES ONTO THE TEMPORARY STATION LINK MEMBERSHIP LIST
```

```
      PROJECT: AGT                    MEMBER: SMDETR                  DATE:
      LIBRARY: DSM                    LEVEL:  01.02                   TIME:
      TYPE:    PCL                    USERID: P326507                 PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    3    ENDDO
    3    EQUATE THE TAIL OF THE STATION LINK MEMBERSHIP LIST TO THE
    3    TAIL OF THE TEMPORARY STATION LINK MEMBERSHIP LIST
    1  ENDPROC
```

```
     PROJECT: AGT                    MEMBER: SMDIVF                    DATE
     LIBRARY: DCM                    LEVEL:  01.09                     TIME
     TYPE:    PL1                    USERID: CI20496                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC:SMDIVF(V#)
   3    <ORDER THE SL'S DOWNSTREAM OF V#'S CURRENT SL SUCH THAT:
   5      - THE PRIMARY CHOICE IS FIRST ON THE LIST
   6      - ALTERNATE CHOICES ARE 2ND, 3RD, ETC.
   8      - INFEASIBLE SL'S ARE ELIMINATED>
   3    <SELECT THE DIVERGE FUNCTION BASED ON SLDIVC(VCURR(V#)) AND
   4     RETURN THE ORDERED AND SHORTENED LIST IN ASSLST>
   3    <USE: SMDIVS - TO SEARCH FOR A LINK OF A GIVEN TYPE
   5      AND SMDIVO - TO ORDER A SET OF LINKS BY OCC OR POCC>

   3    CASEENTRY(SLDIVC(VCURR(V#)))

   3    CASE(1)  <STATION ENTRY>
  12             <ALL SL'S OTHER THAN THE FOLLOWING WILL BE IGNORED BY CASE 1
  14               1 IR - INPUT RAMP
  14               1 BP - BYPASS LINK>
  15              <SIMULATION WILL TERMINATE IF THESE ARE NOT FOUND>
  13              <IN THE CASE OF AN ON-LINE STATION THIS TYPE OF DIVERGE
  15                SHOULD NOT BE USED>
   5      IF     <NEXT STOP OF V# IS THIS STATION>
   7        VNXSTN(V#)=STSIM
   5      THEN
   7        ASSLST:=IR,BP
   5      ELSE
   7        ASSLST:=BP
   5      ENDIF

   3    CASE(2) <DIVERGE TO DOCK, STORAGE, OR MODAL EXIT BEFORE PROCESSING>
  11             <ALL SL'S OTHER THAN THE FOLLOWING WILL BE IGNORED BY CASE 2:
  13               1-N IQ OR D - INPUT QUEUE OR DEBOARD/BOARD
  13               0-1 IS - INPUT TO STORAGE
  15               0-1 MOB - MODAL OUTPUT BEFORE PROCESSING>
  11             <SIMULATION WILL TERMINATE IF THESE ARE NOT FOUND WHEN NEEDED
   5      IF    <THE SINK OF V# IS MODAL OUTPUT BEFORE PROCESSING
   7        VSINK(V#)=MOB
   5      THEN
   7        ASSLST:=MOB
   5      ELSE
   7        IF
   9          VDS(V#)=0           <V# MARKED TO GO TO DOCK>
   7        THEN
   9          ASSLST:=IQ'S OR D'S IN ORDER OF MIN OCC OR POCC
   7        ELSE
   9          ASSLST:=IS,(IQ'S OR D'S IN ORDER OF MIN OCC OR POCC)
   7        ENDIF
   5      ENDIF

   3    CASE(3) <DIVERGE TO OUTPUT,STORAGE, MODAL OUTPUT AFTER PROCESSING>
  11             <ALL SL'S OTHER THAN THE FOLLOWING WILL BE IGNORED BY CASE 3:
  13               1-N OQ - OUTPUT QUEUE OR 1 OR - OUTPUT RAMP
```

```
      PROJECT: AGT                          MEMBER: SMDIVF                          DATE:
      LIBRARY: DSM                          LEVEL:  01.09                           TIME:
      TYPE:    PLL                          USERID: C120496                         PAGE:
START
 COL  ---+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  10              0=1 DS - 0 TO STORE
  10              0=1 MOA - MODAL OUTPUT AFTER PROCESSING>
  11            <SIMULATION WILL TERMINATE IF THESE ARE NOT FOUND WHEN NEEDED
   6       IF     <SINK OF V* IS MODAL OUTPUT AFTER PROCESSING(MOA)>
   7         VSINK(V*)=MOA
   8       THEN
   7         ASSLST:=MOA
   6       ELSE
   7         IF     <V* MARKED TO GO TO STORE BY EVM>
   9           VDS(V*)=1
   7         THEN
   9           ASSLST:=DS,(OG'S IN ORDER OF MIN OCC OR POCC) OR OUTPUT RAMP
   7         ELSE
   9           ASSLST:=(OG'S IN ORDER OF MIN OCC OR POCC) OR OUTPUT RAMP
   7         ENDIF
   6       ENDIF

   3     CASE(4) <DIVERGE OUT OF STORAGE TO INPUT OR OUTPUT>
  11             <ALL SL'S OTHER THAN THE FOLLOWING WILL BE IGNORED BY CASE 3:
  10               1 SI - STORAGE TO INPUT
  10               1 SO - STORAGE TO OUTPUT>
  11             <SIMULATION WILL TERMINATE IF THESE ARE NOT FOUND WHEN NEEDED
   6       IF   <V* MARKED TO GO TO DOCK>
   7         VDS(V*)=0
   8       THEN
   7         ASSLST:=SI
   6       ELSE
   7         ASSLST:=SO
   6       ENDIF

   3     CASE(5) ORDER LIST OF DOWNSTREAM SL'S BY OCC

   3     CASE(6) ORDER LIST OF DOWNSTREAM SL'S BY POCC

   3     ENDCASE

   4   ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SMDIVD                    DATE
      LIBRARY: DSM                    LEVEL:  01.00                     TIME
      TYPE:    PCL                    USERID: CI20496                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  PROC:SMDIVD(L,I) <ORDER A SET OF LINKS BY OCCUPANCY OR PSEUDO-OCC>
    2    FIND LENGTH OF LIST
    3    CASEENTRY(I)     <ORDER METHOD>
    4    CASE(LOCCUPANCY)
    5      DO BUBBLE SORT USING SLOCC OF EACH SL IN L
    3    CASE(PSEUDO-OCCUPANCY)
    5      DO BUBBLE SORT USING SLPOCC OF EACH SL IN L
    6    ENDCASE
    1  ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SMDIVS                    DATE
      LIBRARY: DSM                    LEVEL:  01.01                     TIME
      TYPE:    PLI                    USERID: C120496                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC SMDIVS(V,TYPE,ARMED,SLN)
    3    <BUILD A LIST IN SLN OF ALL SL DOWNSTREAM OF THE CURRENT SL OF
    4     V WHOSE SLTYPE = 'TYPE' REQUESTED IN CALL  AND  IF ARMED
    4     AND NO LINK OF GIVEN TYPE FOUND ABORT RUN>
    6    SLN(1)=0
    3    WHILE
    5      THERE IS ANOTHER DOWNSTREAM SL TO TEST
    3    DO
    5      IF
    7        SLTYPE OF SL = TYPE REQUESTED IN CALL
    5      THEN
    7        ADD SL TO LIST 'SLN'
    5      ENDIF
    3    ENDDO
    3    IF
    5      LIST STILL NULL AND ARMED TO ABORT
    3    THEN
    5      CALL ERROR
    3    ENDIF
    1  ENDPROC
```

```
    PROJECT: AGT                    MEMBER: SMENTR                    DATE
    LIBRARY: DSM                    LEVEL:  01.05                     TIME
    TYPE:    PLL                    USERID: P326507                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

   1  PROC:SMENTR (V)   <ENTRAIN VEHICLES BEHIND V TO V AFTER LAUNCH>
   2    V MUST BE THE LEAD VEHICLE ON THE LINK
   3    V MUST NOT BE QUEUED
   4    IF
   5      V IS NOT ALONE ON THE SL MEMBERSHIP LIST
   6      THE LENGTH OF V IS LESS THAN THE MAXIMUM TRAIN LENGTH
   7    THEN
   8      CREATE A TEMPORARY TAIL FOR THE MEMBERSHIP LIST
   9      DQUE THE V FROM THE LINK MEMBERSHIP LIST
  10      NQUE V ONTO THE TEMPORARY LINK MEMBERSHIP LIST
  11      IF
  12        V IS NOT A TRAIN
  13      THEN
  14        SET ITS CHAIN WORD
  15      ENDIF
  16      FIND THE TAIL OF THE EXISTING TRAIN
  17      AHEAD = TEMP TAIL OF THE TRAIN
  18      WHILE
  19        THE NEXT STOP MATCHES
  20        THE TRAIN LENGTH MAX IS NOT VIOLATED
  21        OTHER VEHICLES EXIST ON THE LINK
  22        OTHER VEHICLES BEHIND V ARE QUEUED FOR REASONS 2 OR 3
  23      DO
  24        SET &VEVCH(LEAD VEHICLE) = XMX + 2 TO SIGNAL TO STATISTICS
  25          COLLECTION THAT THE FOLLOWING VEHICLES IN THE TRAIN HAVE
  26          COME FROM THE QUEUED STATE WHILE THE LEAD VEHICLE HAS
  27          COME FROM THE FEL
  28        REMOVE VEHICLE FROM LINK MEMBERSHIP LIST
  29        PUT VEHICLE ON TEMPORARY TRAIN CHAIN
  30        ADD 1 TO THE LEAD VEHICLE'S TRAIN LENGTH
  31      ENDDO
  32      WHILE
  33        VEHICLES REMAIN ON THE LINK
  34      DO
  35        REMOVE FROM THE LINK MEMBERSHIP LIST
  36        PUT ON THE TEMPORARY MEMBERSHIP LIST
  37      ENDDO
  38      MEMBERSHIP LINK TAIL:= TEMPORARY LINK TAIL
  39    ENDIF
  40  ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SMEVM                    DATE:
      LIBRARY: USM                    LEVEL:  01.06                    TIME:
      TYPE:    PDL                    USERID: P326507                  PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SMEVM(V) <INVOLKED FOR NON-SCHEDULED EMPTY VEHICLES AFTER BOARDING
    2    VDIVST (V):= 1 <INITIALIZE VEHICLE TO GO TO STORAGE>
    3    IF
    4      PVEPR=1 OR <POLICY TO SEND ALL EMPTY VEHICLES OUT OF STATION>
    5      SLOCC(SLSTOR) >= SLCAP(SLSTOR) OR <STORAGE LINK AT CAPACITY>
    6      SMAGEL (P.NEEDD) = 1   <THERE IS A SIMULATED NEED FOR V AT
    7        ANOTHER STATION>
    8    THEN  <VEHICLE WILL NOT DIVERT TO STORAGE>
    9      VDIVST(V):=0
   10    ENDIF
   11  ENDPROC
```

```
     PROJECT: ACT                    MEMBER: SMGDIP4                  DATE
     LIBRARY: DSM                    LEVEL:  01.01                    TIME
     TYPE:    PDL                    USERID: C120496                  PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC:GDIP4
   3    OUTLINES THE INPUT COMMON AREAS
   5      SCISYS
   5      SCISL
   5      SCITL
   5      SCIMAX
   5      SCIFEL
   3    THAT CAN BE UPDATED BY GDIP INPUT
   3    DESCRIPTION GIVES LENGTH:
   5      F - 4 BYTES
   5      H - 2 BYTES
   5      X - 1 BYTE
   3    AND DIMENSIONS OF VARIABLES IN THESE COMMON AREAS
   1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SMLTIM                    DATE:
      LIBRARY: DSM                    LEVEL:  01.14                     TIME:
      TYPE:    PCL                    USERID: P326507                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  LOCAL VAR NAME  DIM    DESCRIPTION
   1  --------------- ------ ------------------------------------------------
   1  B2              -      DELAY TIME DUE TO SCHEDULE
   1  NXT             -      NEXT DESIRED LAUNCH TIME
   1  ------------------------------------------------------------------------
   1  PROC:SMLTIM(V) <COMPUTE SCHEDULE DELAY COMPONENT OF BOARD TIME FOR
   3    THE SCHEDULED SERVICE POLICY>
   3    IF
   5      PVSPAC=2      <VEHICLES DEPART USING FIXED DEPARTURE TIMES>
   3    THEN
   5      R:=VROUTE(V)
   5      NXT:=PLSCHT(R)+PRTEHW(R)    <NEXT DESIRED LAUNCH TIME = LAST
   5      SCHEDULED LEAVE TIME PLUS DESIRED HEADWAY TIME>
   5      <DETERMINE SCHEDULE DELAY>
   5      IF <DESIRED NEXT LAUNCH TIME HAS PASSSED ALREADY>
   7        CLOCK > NXT
   5      THEN
   7        B2:=0
   5      ELSE <DESIRED NEXT LAUNCH TIME HAS NOT YET PASSED>
   7        B2:= NXT - CLOCK
   5      ENDIF
   5      PLSCHT(R):=NXT <UPDATE LAST LAUNCH TIME>
   3    ELSE <VEHICLES DEPARTURE IS DETERMINED USING THE MIDWAY POINT
   5      METHOD>
   5      NXT:=(PNXSLV(R)+PLSTLV(R)+PRTEHW(R))/2
  34                                    <SET NEXT LAUNCH TIME TO AVERAGE OF
  31                                   THE TIME THE NEXT SHOULD LEAVE(AFTER
  31                                   THIS ONE) AND THE TIME THE LAST DID
  34                                   LEAVE(THE ONE BEFORE THIS ONE>
   5      <DETERMINE SCHEDULE DELAY>
   5      IF <DESIRED NEXT LAUNCH TIME HAS PASSED ALREADY>
   7        CLOCK > NXT
   5      THEN
   7        B2:=0
   5      ELSE <DESIRED NEXT LAUNCH TIME HAS NOT YET PASSED>
   7        B2:= NXT - CLOCK
   5      ENDIF
   5      UPDATE PLSTLV(R) AFTER OTHER DELAYS ARE FOUND
   5      PNXSLV(R):=PNXSLV(R)+PRTEHW(R)
   3    ENDIF

   1  ENDPROC
```

```
    PROJECT: AGT                    MEMBER: SMNXST                      DATE
    LIBRARY: GSM                    LEVEL:  01.06                       TIME
    TYPE:    PDL                    USERID: P326507                     PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

  1  PROC:SMNXST(V)            <SET THE NEXT STOP OF V>
  3    CASEENTRY(POLSER)       <SERVICE POLICY>

  3    CASE(DEMAND RESPONSIVE SINGLE PARTY)
  5      IF
  7        VTRIPQ(V)=0         <V EMPTY>
  5      THEN
  7        RUN SMEVM(V)        <RUN EMPTY VEHICLE MANAGEMENT>
  5      ELSE                  <V IS NOT EMPTY>
  6        VDIVST(V)=0         <MARK V NOT TO DIVERT TO STORAGE>
  6        VNXSTN(V)=IDEST(VTRIPQ(V))  <SET NEXT STOP TO DEST OF TRIP ON V>
  4      ENDIF

  3    CASE(DEMAND RESPONSIVE MULTIPARTY)
  5      IF
  7        VTRIPQ(V)=0         <V EMPTY>
  5      THEN
  7        RUN SMEVM(V)        <RUN EMPTY VEHICLE MANAGEMENT>
  5      ELSE                  <V IS NOT EMPTY>
  6        VDIVST(V)=0         <MARK V NOT TO DIVERT TO STORAGE>
  6        VNXSTN(V)=IDEST(HDEQCH(VTRIPQ(V)))  (NEXT STOP=DEST OF FIRST TRIP
  6        ON V>
  4      ENDIF

  3    CASE(SCHEDULED)
  4      VNXSTN(V):=0          <NOTE: THE NEXT STOP DOES NOT MATTER IN THE CAS
 32                                 OF SCHEDULED SERVICE,SINCE ONLY FULL
 32                                 TRAINS PASS THROUGH THE STATION AND NO
 32                                 ACTIVE ENTRAIN/DETRAIN OPERATION IS
 32                                 PERFORMED IN THE STATION>

  3    ENDCASE
  1  ENDPROC
```

```
      PROJECT: AGT                          MEMBER: SMRNG                          DATE
      LIBRARY: CSM                          LEVEL:  01.04                          TIME
      TYPE:    PDL                          USERID: D425537                        PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC RNG (SEED,RANDOM NUMBER) <RANDOM NUMBER GENERATOR>
    3    KSEED:=KSEED*65539
    3    IF
    5      KSEED<0
    3    THEN
    5      KSEED:=KSEED+2147483647+1
    3    ENDIF
    3    RANDNO:=KSEED
    3    RANDNO:=RANDNO*.4656613E-9
    1  PROCEND
```

```
      PROJECT: AGT                    MEMBER: SMRSEL                    DATE
      LIBRARY: DSM                    LEVEL:  01.02                     TIME
      TYPE:    PLL                    USERID: D425537                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  PROC: RANDSEL(USERDIST,STRT,END,SEED,I)
    3    USERDIST    K    ANY USER PROBABILITY DISTRIBUTION IN THE FORM OF A
   20                      CUMMULATIVE DISTRIBUTION FUNTION (PUT IN THIS FORM
   20                      BY THE INPUT PROCESSOR) OF K ELEMENTS.  (I.E.,
   20                      USERDIST(K)=1.0 ALWAYS)
    3    (STRT-END+1)-  NUMBER OF ELEMENTS IN USERDIST
    3    SEED           -  RANDOM NUMBER SEED
    3    I              -  VALUE FROM 1 TO K THAT WAS RANDOMLY SELECTED
    1  ------------------------------------------------------------------
    3    CALL SMRNG FOR RANDOM NUMBER
    3    I:=0
    3    DO
    5      I:=I+1
    3    WHILE
    5      USERDIST(I) IS LESS THAN RANDOM NUMBER AND
    5      END OF DISTRIBUTION IS NOT REACHED
    3    ENDDO
    3    SET ERROR RETURN IF ENTRY IS NOT FOUND
    1  ENDPROC
```

```
      PROJECT: AUT                    MEMBER: SMTABQ                    DATE
      LIBRARY: DSM                    LEVEL:  01.26                     TIME
      TYPE:    PDL                    USERID: P32650/                   PAGE
START
 CDL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SMTABQ(T)  <TRIP JOINS BOARDING QUEUE>

    1  ------------------------------------------------------------------
    1  LOCAL VAR NAME  DIM    DESCRIPTION
    1  --------------- ------ ---------------------------------------------
    1  FOUND           -      INDICATES A VEHICLE CAN SERVICE THE TRIP BEING
   26                         PROCESSED, HOWEVER,IN SCHEDULED SERVICE,
   26                         DESTINATION COMPATIBILITY MUST ALSO BE PROVEN
   26                         BEFORE THE TRIP CAN BOARD
   26                         T===>FOUND
   26                         F===>NONE FOUND
    1  MATCH    _      -      INDICATES WHETHER OR NOT THE VEH & TRIP ARE
   26                         COMPATABLE
   26                         1===>MATCH
   26                         2===>NO MATCH
    1  ------------------------------------------------------------------
    3    NQ T INTO SEQ1L <THE BOARDING QUEUE FOR ITS STATION>
    3    COLLECT STATISTICS FOR TRIPS ARRIVING AT THE BOARDING QUEUE
    3    COLLECT STATISTICS FOR PASSENGERS ARRIVING AT THE BOARDING QUEUE
    3    COLLECT STATISTICS FOR TRIPS ENTERING THE QUEUED STATE

    3    FOUND:=FALSE       <INITIALIZE FOUND PRIOR TO LOCATING A VEHICLE>

    3    CASEENTRY (HOLSER)

    5      CASE (1,2)       <SERVICE POLICY IS SINGLE PARTY OR MULTIPARTY
    5      DEMAND RESPONSIVE>

    6      I:=1 <INITIALIZE THE PRIORITY LIST INDEX>
    6      DO

    7        CASEENTRY(PVSER(I)) <USING A PRIORITIZED SEARCH OF VEHICLE
    7        SOURCES, CHECK SOURCES IN ORDER>

    7        CASE(1)-                    <LOCAL STORAGE>
    7          FOR
   11            EACH VEHICLE(V) ON SLSTOR
    9          DO     <SEARCH FOR AN AVAILABLE EMPTY>
   11            IF <V IS AVAILABLE>
   14               VQREAS(V)=4
   11            THEN <UNSTORE V>
   13              FOUND:=T
   13              COLLECT STATISTICS FOR VEHICLES GOTTEN FROM LOCAL STORAGE
   13              VDIVST(V):=0 <MARK V TO GO TO DOCK>
   13              IF <V AT HEAD OF SLSTOR>
   15                VMEMCH(SLMEMT(SLSTOR))=V
   13              THEN <GET V MOVING>
   15                VQREAS(V):=1
   15                RUN SCPMAC(VSL(V),SELF)
   13              ELSE <V NOT AT HEAD OF SL>
```

```
      PROJECT: AGT                    MEMBER: SMTABQ                    DATE
      LIBRARY: DSM                    LEVEL:  01.25                     TIME
      TYPE:    PLL                    USERID: P3C6607                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  15                VGREAD(V):=2 <MARK V AS READY TO MOVE>
  13              ENDIF
  13              <NOTE: SETTING VGREAS(V) HERE TO 1 OR 2 WILL CAUSE
  20                     ANY ACTIVITY THAT WAS DESIGNATED TO OCCUR ON
  20                     SLSTOR AFTER THE STORE OPERATION TO BE BYPASSED.
  11            ENDIF
   9          UNTIL
  11            FOUND = 1 OR
  11            ALL VEHICLES ON SLSTOR HAVE BEEN SEARCHED
   9          ENDDO

   7        CASE(2)                  <FETCH AN EMPTY FROM ELSEWHERE IN NET>
   9          FOUND:=T
   9          GET V <A FREE XTN TO REPRESENT THIS ARRIVING EMPTY VEHICLE>
   9          RUN SMTABQ1 <INITIALIZE V AND SET VTIME(V):= -1>
   9          T:=PICK USER'S EMPTY VEHICLE INTERARRIVAL TIME DIST
   9          SCHED(V,T)
   9          UPDATE STATS FOR THE VEHICLE'S ARRIVAL AT STATION
   9          COLLECT STATISTICS FOR VEHICLES GOTTEN FROM ELSEWHERE IN THE
 -11            NETWORK

   7        CASE(3)                  <LOOK UPSTREAM OF THE DOCK>
   9          FOR
  11            EACH USER SPECIFIED SL IN THE LIST PSLIST(SL)
   9          DO
  11            FOR
  13              EACH VEHICLE(V) ON EACH TRAIN ON THE SL
  11            DO
  15              IF
  15                VTRIPQ(V)=0 & <V EMPTY>
  15                VRES(V)=F <V NOT RESERVED>
  13              THEN <MARK VEHICLE AS RESERVED>
  15                VRES(V):=T
  15                FOUND:=T
  15                COLLECT STATISTICS FOR VEHICLES GOTTEN FROM UPSTREAM
  17                  OF THE STATION LINK
  13              ENDIF
  11            UNTIL
  13              FOUND=T OR
  13              ALL VEHICLES ON THE SL HAVE BEEN TRIED
  11            ENDDO
   9          UNTIL
  11            FOUND = T OR
  11            ALL STATION LINKS HAVE BEEN TRIED
   9          ENDDO
   7        ENDCASE
   5      UNTIL
   7        FOUND = T OR
   7        1 = KNSVP <ALL SEARCH PRIORITIES HAVE BEEN TRIED>
   5      ENDDO
```

```
      PROJECT: AGT                    MEMBER: SMTABQ                    DATE
      LIBRARY: USM                    LEVEL:  01.28                     TIME
      TYPE:    PCL                    USERID: P326507                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

   6      IF <ALL REQUESTED METHODS FAIL>
   7        FOUND = F
   5      THEN
   7        FOUND = T
   7        GET V <A FRESH XIN TO REPRESENT THIS ARRIVING EMPTY VEHICLE>
   7        RUN SMTABQ1 <INITIALIZE V AND SET VTIME(V):= -1>
   7        T:=PICK USER'S EMPTY VEHICLE INTERARRIVAL TIME DIST
   7        SCHED (V,T)
   7        UPDATE STATS FOR THE VEHICLE'S ARRIVAL AT STATION
   7        COLLECT STATISTICS FOR VEHICLES ARRIVING FROM ELSEWHERE IN
   9          THE NETWORK
   5      ENDIF

   3    CASE (3) <SERVICE POLICY IS SCHEDULED>
   5      RUN SMTABQ2
   3    ENDCASE
   1  ENDPROC
```

```
    PROJECT: AGT                    MEMBER: SMTABQ2
    LIBRARY: DSM                    LEVEL:  01.01
    TYPE:    PDL                    USERID: C120496
PART
LOL ----+----1----+----2----+----3----+----4----+----5----+---
      WHILE                                                                  000100
        FOUND = F AND AT LEAST ONE VEHICLE/TRAIN IS IN THE BOARDING OR       000200
        JOINT EVENT                                                          000300
    5 FOR                                                                    000400
    7   EACH LEAD VEHICLE                                                    000500
    5 DO                                                                     000600
    7   WHILE                                                                000700
    9     FOUND = F                                                          000800
    7   FOR                                                                  000900
    9     EACH VEHICLE IN THE TRAIN                                          001000
    7   DO                                                                   001100
    9     IF   <T CAN FIT ON VEHICLE>                                        001200
   11       VCAP >= VNPASS(V) + TPASS(T)                                     001300
    9     THEN                                                               001400
   11       FOUND:= TRUE                                                    001500
   11       CASEENTRY(COMPATIBILITY METHOD)                                 001600
   11       CASE(PROBABILITY OF COMPATIBILITY)                              001700
   13         MATCH:=SMRSEL(PCOMPD)                                         001800
   11       CASE(TABLE OF STATIONS FOR EACH ROUTE)                          001900
   13         IF                                                            002000
   15           THE ROUTE ASSIGNMENT TABLE IS USED                          002100
   13         THEN                                                          002200
   15           IF                                                          002300
   17             THE ROUTE OF THE VEHICLE MATCHES THE ROUTE                002400
   17             ASSIGNED TO THE TRIP'S DESTINATION                        002500
   15           THEN                                                        002600
   17             MATCH = 1                                                 002700
   15           ELSE                                                        002800
   17             MATCH = 2                                                 002900
   15           ENDIF                                                       003000
   13         ELSE                                                          003100
   15           IF                                                          003200
   17             DEST OF T IS ON THE ROUTE OF V                            003300
   15           THEN                                                        003400
                  MATCH:=1                                                  003500
                ELSE                                                        003600
                  MATCH:=2                                                  003700
   15           ENDIF                                                       003800
   13         ENDIF                                                         003900
   11       ENDCASE                                                         004000
   11       IF     <COMPATIBLE>                                             004100
   13         MATCH=1                                                       004200
   11       THEN                                                            004300
   13         DQ T FROM SBQTL <REMOVE T FROM BOARDING QUEUE>                004400
   13         NQ T INTO VBLTL(V) <INSERT T INTO BOARDING LIST OF V>         004500
   13         VNPASS(V) = VNPASS(V) + TPASS(T) <INCREMENT BOARDING PASS>    004600
   13         IF (WRITING TRIP'S VEH EVENT FILE)                            004700
   15           WRITE TVF RCD FOR LATE ARRIVING TRIPS                       004800
   13         ENDIF                                                         004900
   11       ENDIF                                                           005000
    9     ENDIF                                                            005100
    7   ENDDO                                                              005200
    5 ENDDO                                                                005300
```

```
      PROJECT: AGT                    MEMBER: SODATA                    DATE:
      LIBRARY: DSM                    LEVEL:  01.00                     TIME:
      TYPE:    PLL                    USERID: C120496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SODATA    <INITIALIZE TABLES>
    3    INITIALIZE TABLES USED IN OR WITH BLOCK DATA SUBPROGRAMS
    1  ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SOMTIX                    DATE
      LIBRARY: CSM                    LEVEL:  00.01                     TIME
      TYPE:    PLL                    USERID: H639547                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC SOMTIX <SAVE PARM FIELD ADDRESS & INVOKE OP CONTROL>
    3    STORE PARM FIELD ADDRESS UPON INITIAL ENTRY
    4    RUN SOUTPT
    3    <ENTRY POINT FOR RETRIEVING PARM FIELD & INVOKING INDEX UPDATE
    5      PROCESSING>
    3    ENTRY SOUPTX <UPDATE INDEX FILE>
    5      LOAD PARM FIELD ADDRESS
    5      RUN SOMTIX,<WRITE INDEX FILE ENTRY>
    1  END PROC
```

```
     PROJECT: AGT                        MEMBER: SOPSUM
     LIBRARY: DSM                        LEVEL:  01.01
     TYPE:    PDL                        USERID: C120496
PART
COL  ----+----1----+----2----+----3----+----4----+----5----+----6.

     PROC SOPSUM  <PERFORMANCE SUMMARY PROCESSING>                          000100
       COMPUTE RATIOS, PERCENTAGES, AVERAGES PER HOUR FROM SUMS ACCUMULATED  000200
 3       AT READ TIME FROM MISCELLANEOUS STATISTICS                          000300
 3     RECALCULATE AVERAGE TIMES USING PSUMM(225-316)                        000400
 3     WRITE PERFORMANCE SUMMARY FILE                                        000500
 3     PRINT REPORT                                                          000600
 1   ENDPROC                                                                 000700
```

```
     PROJECT: ACT                    MEMBER: SOUTPT                    DATE
     LIBRARY: PSM                    LEVEL:  00.01                     TIME
     TYPE:    PLL                    USERID: H639547                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC SOUTPT <OUTPUT PROCESSOR CONTROL>
   6       <BASIC OUTPUT PROGRAM TO OBTAIN & LIST DATA FROM THE RAW STATISTI
   7        FILE IN RESPONSE TO USER OUTPUT REQUEST COMMANDS> DATA REQUESTS
   7        ARE ACCUMULATED UNTIL A READ (DATA ACQUISITION) COMMAND IS EN-
   7        COUNTERED. AT THIS POINT THE DATA REQUESTS ARE SERVICE & THE
   7        REQUESTED DATA DISPLAYS ARE PRODUCED.>
   3    RUN SUZNIT <PROGRAM INITIALIZATION>
   3    WHILE  <USER REQUESTS REMAIN>
   5      ¬ENDFILE
   3    DO <PROCESS REQUEST >
   5      READ INPUT REQUEST <USER COMMAND>
   7        CASEENTRY(TYPE) <REQUEST TYPE>
   9          CASE(1) <WANT (DATA REQUEST>
  11            READ REQUEST DATA
  11            FOR <NUMBER OF FORMATS AVAILABLE>
  13              MTYPE=1 TO KMF
  11            DO <VALIDATE FORMAT TYPE & DETERMINE NUMERIC IDENTIFIER>
  13              IF
  15                NAME=FORM(MTYPE)
  13              THEN                        CARD TO LIST OF VALID
  15                NAME:=0
  13              ELSE
  15                IF <END OF LIST>
  17                  MTYPE=KMF
  15                THEN <ERROR EXISTS- INVALID FORMAT>
  17                  ERROR <INVALID FORMAT>
  15                ENDIF
  13              ENDIF
  11            UNTIL
  13              NAME=0
  11            ENDDO
  11            IF <NO ERROR>
  13              ¬ERROR
  11            THEN <PERFORM REQUEST FILING>
  13              RUN ZREQU <FILE REQUEST & OBTAIN STORAGE ASSIGNMENT>
  13              BIN:=BIN+1 <NEXT AVAILABLE BIN ID>
   7          CASE(2)   <READ DATA FILE FOR PRECEEDING REQUESTS>
   9            READ ACCUMULATION INTERVAL REQUIRED (MIN)
   9            RUN ZREAD <READ RAW STATS FOR REQUESTED DATA>
   9            FOR <ALL REQUESTS FILED>
  11              REQ=1 TO IREQ
   9            DO <PRODUCE REQUIRED DATA DISPLAY>
  11              BINR:=REQTBL(IREQ,1) <BIN NUMBER ASSIGNED TO REQUEST>
  11              FORM:=BPTR(BIN)+2 <FORMAT ID IN BIN>
  11              CASEENTRY(FORM) <DISPLAY FORMAT REQUIRED>
  13                CASE(1) <HISTOGRAM>
  15                  RUN ZHIST
  13                CASE(2) <TIME SERIES PLOT>
  15                  RUN ZPLOT
  13                CASE(3,4) <LIST OR SUMMARY>
```

```
      PROJECT: AGT                    MEMBER: SOUTPT                    DATE:
      LIBRARY: USM                    LEVEL:  00.01                     TIME:
      TYPE:    PEL                    USERID: H69547                    PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  15                  RUN ZLIST
  11              ENDCASE
   9            ENDDO
   8            WIN:=1
   7          ENDDO
   5        ENDCASE
   3      ENDDO
   1  ENDPROC
```

```
    PROJECT: AGT                        MEMBER: SOWTIX
    LIBRARY: DSM                        LEVEL:  00.01
    TYPE:    PDL                        USERID: C120496
:ART
:OL  ----+----1----+----2----+----3----+----4----+----5----+--

     PROC SOWTIX <DECODE PARM FIELD INFO & UPDATE INDEX>                000100
       PARSE PARM LIST                                                  000200
       WRITE NAME OF LOAD MODULE, DATE & TIME                           000300
   1 ENTRY SOWTIW                                                       000400
   3   LIST FILES USED IN INDEX                                         000500
   1 ENTRY SOWTIY                                                       000600
   3   WRITE PERSUM MEMBER NAME IN PERSUM MEMBER                        000700
   1 ENDPROC                                                            000800
```

```
PROJECT: AGT                    MEMBER: SOZNIT                    DATE:
LIBRARY: SSM                    LEVEL:  00.01                     TIME:
TYPE:    PLL                    USERID: H639547                   PAGE:
START
 COL  ---+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC SOZNIT <DETAILED STATION MODEL OUTPUT PROCESSOR INITIALIZATION>
   2    ESTABLISH LINKAGE TO INTERRUPT HANDLER
   3    INITIALIZE SYSTEM DATA
   4    INITIALIZE MESSAGES
   7    FOR
   8      I=1,12
   3    DO <INITIALIZE REQUEST TABLE>
   5      ZREQUE(I,1)=0
   3    ENDDO
   3    RUN OBIN <PERFORM INITIAL BIN ALLOCATIONS>
   3    DO <PERFORM DUMMY READ TO OBTAIN SIMULATION CHARACTERISTICS>
   5      RUN ZREQU <FILE DUMMY DATA REQUEST>
   5      RUN ZREAD <READ SYSTEM CHARACTERISTICS FROM RAW STATISTICS FILE>
   3    ENDDO
   1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SREAD02                    DATE:
      LIBRARY: DSM                    LEVEL:  00.01                      TIME:
      TYPE:    PDL                    USERID: H639547                    PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC READ02 <READ TYPE 2 FOLLOWERS - SYSTEM STATISTICS>
    4     <ITEMS NOT RELATED TO STATION STATES>
    3    READ FOLLOWER RECORD
    4     <ITEMS RELATED TO STATE 1 - IN STATION>
    3    READ FOLLOWER RECORD
    4     <ITEMS RELATED TO STATE 2 - BOARDING>
    3    READ FOLLOWER RECORD
    4     <ITEMS RELATED TO STATE 3 - DEBOARDING>
    3    READ FOLLOWER RECORD
    4     <ITEMS RELATED TO STATE 4 - LAUNCH>
    3    READ FOLLOWER RECORD
    3    CUSAM=.SIZE/60. <CLOCK UNITS/SECOND>
    3    FOR
    5      EACH REQUEST REQUIRING TYPE 2 DATA
    3    DO
    5      ISUB= ZREQUE(9,IREQ) <SUB CATAGORY NUMBER>
    5      IF
    7        ISUB<=8
    5      THEN <STATISTIC RELATED TO OVERALL STAION PERFORMANCE>
    9          VAL = STAT(ISUB)
    7        ELSE <DETERMINE POSITION OF DATA IN RECORD>
    9          ISUB=ISUB-8
    9          ISUB1=ISUB/27
    9          ISUB2=ISUB-(ISUB1*27)
    9          IF
    1            ISUB2>15
    9          THEN <REQUIRED VALUE IS HALFWORD>
   11            VAL=HSTAT(ISUB1*12+ISUB2-15)
    9          ELSE
   11            IF
   13              ISUB2>6
   11            THEN <REQUIRED VALUE IS A FULLWORD>
   13              VAL=FSTAT(ISUB1*9+ISUB2-6)/CUSAM
   11            ELSE <REQUIRED VALUE IS REAL>
   13              VAL=RSTAT(ISUB1*6+ISUB2)
   13              IF
   15                ISUB2>3
   13              THEN <VALUE MUST BE CONVERTED TO SECONDS>
   15                VAL=VAL/CUSAM
   13              ENDIF
   11            ENDIF
    9          ENDIF
    7        ENDIF
    5      ENDIF
    5      RUN STORE <STORE THE DATA VALUE IN A BIN>
    3    ENDDO
    1  ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SREAD03                    DATE:
      LIBRARY: DSM                    LEVEL:  01.00                      TIME:
      TYPE:    PGL                    USERID: H639547                    PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC READ03 <READ TYPE 3 FOLLOWERS - STATION LINK STATISTICS>
    3    FOR
    5      SL=1,<NSL>
    3    DO
    6      READ ITEMS RELATED TO STATION LINK OCCUPANCY,
   11           ITEMS RELATED TO STATION LINK PROCESSING,
   11           ITEMS RELATED TO STATION LINK QUEUING
    3    ENDDO
    3    CUSAM= CSIZE/60 <CLOCK UNITS/SECOND>
    3    FOR
    5      EACH REQUEST REQUIRING THIS RECORD TYPE
    3    DO
    5      ISUB= ZREQUE(9,IREQ) <SUB CATAGORY ID>
    7        ISUB1=ISUB/9
    7        ISUB2=ISUB-(ISUB1*9)
    7        SLN=ZREQUE(4,IREQ) <STATION LINK NUMBER>
    7        IF
    9          ISUB2>5
    7        THEN <VALUE IS A HALFWORD STATISTIC>
    9          VAL=HSTAT1((SLN-1)*12+ISUB1*4+ISUB2-5)
    7        ELSE
    9          IF
   11            ISUB2>2
    9          THEN <VALUE IS A FULLWORD STATISTIC REQUIRING TIME
   13               CONVERSION TO SECONDS>
   11            VAL=FSTAT1((SLN-1)*9+ISUB1*3+ISUB2-2)/CUSAM
    9          ELSE <VALUE IS A REAL STATISTIC>
   11            VAL=RSTAT1((SLN-1)*6+ISUB1*2+ISUB2)
   11            IF
   13              ISUB2>1
   11            THEN <VALUE MUST BE CONVERTED TO SECONDS>
   13              VAL=VAL/CUSAM
   11            ENDIF
    9          ENDIF
    7        ENDIF
    5      ENDIF
    5      RUN STORE <STORE THE VALUE IN A BIN>
    3    ENDDO
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SREAL04                    DATE:
      LIBRARY: SSM                    LEVEL:  00.01                      TIME:
      TYPE:    PLL                    USERID: H63547                     PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC SREAL04 <READ TYPE 4 FOLLOWERS - TRIP LINK STATISTICS>
    3    FOR
    5      TL=1,KNTL)
    6    DO
    8      READ ITEMS RELATED TO TRIP LINK OCCUPANCY,
   11           ITEMS RELATED TO TRIP LINK PROCESSING,
   11           ITEMS RELATED TO TRIP LINK QUEUING
    2    ENDDO
    3    CUSAM= CSIZE/60 <CLOCK UNITS/SECOND>
    6    FOR
    5      EACH REQUEST REQUIRING DATA IN THIS RECORD TYPE
    8    DO
    5      ISUB= ZREQUE(4,IREQ) <SUBCATEGORY ID>
    6      ISUB1=ISUB/18
    6      ISUB2=ISUB-(ISUB1*18)
    5      TLN=ZREQUE(4,IREQ) <TRIP LINK NUMBER>
    6      IF
    7        ISUB2>10
    6      THEN <VALUE IS A HALF WORD STATISTIC>
    9          VAL=HSTAT1((TLN-1)*24+ISUB1*8+ISUB2-10)
    8      ELSE
    7        IF
    9          ISUB2>6
    7        THEN <VALUE IS A FULLWORD STATISTIC>
    9          VAL=FSTAT1((TLN-1)*12+ISUB1*4+ISUB2-6)/CUSAM
    9          ELSE <VALUE IS REAL VARIABLE>
   11            VAL=RSTAT1((TLN-1)*18+ISUB1*6+ISUB2)
   11            IF
   13              ISUB2>2
   11            THEN <TIME CONVERSION TO SECONDS REQUIRED>
   13              VAL=VAL/CUSAM
   11            ENDIF
    4          ENDIF
    7        ENDIF
    5      ENDIF
    5      RUN STORE <STORE THE VALUE IN A BIN>
    3    ENDDO
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SREQTLU                    DATE:
      LIBRARY: LSM                    LEVEL:  01.00                      TIME:
      TYPE:    PEL                    USERID: C120496                    PAGE:
START
 COL  ---+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SREQTLU   <RECORD/REQUEST CORRELATION>
    3    FOR
    5      ALL SUBCATEGORIES
    3    DO
    5      FOR
    7        ALL REQUEST TABLE ENTRIES
    5      DO
    7        IF
    9          MATCH FOUND
    7        THEN
    9          CHAIN LIFO WITHIN SAME ENTITY NUMBER
    7        ENDIF
    5      ENDDO
    3    ENDDO
    1  ENDPROC
```

```
      PROJECT: AGI                          MEMBER: SSASAV                        DATE
      LIBRARY: USM                          LEVEL:  01.01                         TIME
      TYPE:    PDL                          USERID: C120496                       PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC SSASAV <INITIALIZE SYSTEM STATUS AREA WORDS>
    3    AT LINKAGE EDIT TIME THIS SERVES TO CAPTURE THE ADDRESSES OF
    4      THE ADDRESSES OF THE START OF THE INPUT COMMONS, START OF THE
    5      MODEL AREA COMMONS AND END OF THE COMMONS
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SSETUP                    DATE:
      LIBRARY: DSM                    LEVEL:  00.01                     TIME:
      TYPE:    PCL                    USERID: H639547                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC SETUP <INTIALIZE MAJOR CATAGORY TABLE>
    2    I=I+1
    3    DO <INITIALIZE TABLE OF MAIN CATAGORIES>
    5      MATTAB(I)=MATTAX(I)
    3      I=I+1
    3    ENDDO
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SSLEAV                    DATE:
      LIBRARY: DSM                    LEVEL:  01.26                     TIME:
      TYPE:    PCL                    USERID: P326507                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SSLEAV(V)
    2    <DONE WHEN V IS ACTUALLY LEAVING, I.E.,HAS PAST PASSED SSTEST>
    3    SL:=VSL(V)
    4    <PROCESS THE LEAVING VEHICLE>
    5    SLOCC(SL):=SLOCC(SL)-VTRLEN(V) <DECREASE OCCUPANCY OF OLD SL BY TRAIN
    6    LENGTH OF V>
    7    DQ V FROM SLMEMT(SL)  <REMOVE V FROM MEMBERSHIP CHAIN OF OLD SL>
    8    COLLECT STATS FOR VEHICLE LEAVING THE STATION LINK
    9    IF      <V IS ONLY VEHICLE OR TRAIN ON SL>
   10      SLMEMT(SL)=0   <NOTE: ONLY THE LEAD VEHICLE OF THE TRAIN IS ON THE
   11      STATION LINK MEMBERSHIP LIST>
   12    THEN
   13      IF  <CURRENT SL HAS DEBOARD/BOARD EVENTS>
   14        SLTYPE(SL) = 2
   15      THEN
   16        SLPOCC(SL):=0           <RESET AVAILABLE BERTHS TO FULL CAP OF SL>
   17      ENDIF
   18    ELSE
   19      NEXTV:=VMEMCH(SLMEMT(SL)) <IDENTIFY THE FOLLOWING VEHICLE/TRAIN>
   20      <PROCESS THE FOLLOWING VEHICLE ON THIS STATION LINK>
   21      CASEENTRY(VGREAS(NEXTV)) <IF NEXTV WAS DELAYED BY V, GET IT
   22      GOING>

   23      CASE(2) <V WAS IN FRONT & NEXTV OTHERWISE DONE>
   24        VGREAS(NEXTV):=1 <NEXTV IS NOW QUEUED DUE TO CONGESTION SO THAT
   25                          SASPRM WILL RECOGNIZE IT AS NOW READY TO
   26                            TRY TO LEAVE>
   27        SSPMAC(SL, S <TRY TO GET NEXTV GOING>

   28      CASE(3) <V WAS IN FRONT & NEXTV WAITING TO START L-EVENT>
   29        <NOTE: VSEVNT(NEXTV) ALREADY SET TO VEHICLE EVENT &
   30                VMEVNT(NEXTV) ALREADY SET TO LAUNCH EVENT>
   31        SSMOD (NEXTV)          <GET NEXTV GOING AGAIN>

   32      CASELSE
   33        <NO OP> <NOTE: CASE=0 ===> NEXTV ON FCL - SO DO NOT TOUCH IT
   34                              =1 ===> NEXTV QUEUED DUE TO CONG/FAILURE ---
   35                                      BUT COULDN'T BE SINCE WASN'T AT
   36                                      TOP OF SL - SO NOT APPLICABLE
   37                              =4 ===> NEXTV IS STORED - SO LEAVE IT>
   38      ENDCASE
   39    ENDIF
   40    <PROCESS VEHICLES ON THE UPSTREAM STATION LINK(S)>
   41    SSPMAC (SL,U)<TRY TO GET VEHICLE(S) ON SL(S) UPSTREAM OF OLD
   42                  SL MOVING, SINCE NEXTVS EXIT MAY HAVE MADE ROOM
   43                  FOR NEW VEHICLE(S)/TRAIN(S)>

   44  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SSMOD                    DATE
      LIBRARY: USM                    LEVEL:  01.16                    TIME
      TYPE:    PDL                    USERID: 9326507                  PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

   1  PROC:SSMOD(V)  <MODEL THE VEHICLE ON ITS CURRENT STATION LINK>
   3    IF
   5      VMEVNT(V)¬=0  AND    <NOT PRIOR TO FIRST EVENT FOR THIS SL>
   5      VWREAS(V)¬=3         <NOT WAITING TO START LAUNCH EVENT BECAUSE
   5      QUEUED DUE TO OTHER VEHICLE IN FRONT AND WAITING TO START
   5      LAUNCH EVENT>
   3    THEN   <PERFORM AFTER-TIME SEGMENT PROCESSING FOR CURRENT EVENT>
   5      RUN SSMODA(V)
   3    ENDIF

   3    ADONES:=F              <INITIZE INDICATOR TO: "THERE ARE MORE
   3    VEHICLE EVENTS TO OCCUR TO V ON ITS SL.">
   3    RUN SSMODN(V)          <DETERMINE NEXT EVENT ON THIS SL OR 'DONE'>
   3    IF
   5      ADONES=F  AND     <NOT DONE WITH ALL EVENTS ON THIS SL AND>
   5      VWREAS(V)=0      </ HAS NOT BEEN PUT IN A QUEUE DURING MODELING>
   3    THEN   <PERFORM BEFORE-TIME SEGMENT PROCESSING FOR NEXT EVENT>
   5      RUN SSMODB(V)
   3    ENDIF

   1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SSMODA                          DATE
      LIBRARY: DSM                    LEVEL:  01.31                           TIME
      TYPE:    PLL                    USERID: PJZ6507                         PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  PROC:SSMODA(V)
    3    <WHAT TO DO AT END OF TIME SEGMENT FOR THESE PROCESSES>
    3    SL:=VSL(V)

    3    CASEENTRY(VMEVNT(V))      <BASED ON THE VEHICLE EVENT FOR WHICH V
   29                               SPENT THIS PAST TIME SEGMENT ON THE PLL>
    3    CASE(HEADWAY ZONE TRAVEL)
    5      SLHZF(SL):=F     <TURN THE HEADWAY ZONE FLAG OFF>
    5      RUN SSMMAC(SL,UPSTREAM)              <TRY TO GET VEHICLE(S) ON SL(S)
   41                                           UPSTREAM OF THE OLD SL MOVING,
   41                                           SINCE THE HEADWAY ZONE IS NOW
   41                                           CLEAR>

    3    CASE(TRAVEL)
    5      <NO OP>

    3    CASE(DESBARD)
    5      INCLUDE SSMODA1

    3    CASE(BOARD)
    5      INCLUDE SSMODA2

    3    CASE(JOINT)
    5      INCLUDE SSMODA3

    3    CASE(STORE)
    5      <NO OP>

    3    CASE(LAUNCH)
    5      COLLECT STATISTICS FOR A VEHICLE FINISHED WITH LAUNCH
    5      IF  <ENTRAINMENT/DETRAINMENT POLICY IN EFFECT>
    7        PENTS = T
    5      THEN
    7        RUN SMENTR (V)
    5      ENDIF

    3    ENDCASE
    1  ENDPROC
```

```
PROJECT: AGT                    MEMBER: SSMODA1
LIBRARY: DSM                    LEVEL:  01.03
TYPE:    PDL                    USERID: C120496
PART
COL  ----+----1----+----2----+----3----+----4----+----5----+----
```

```
  CASE(DEBOARD)                                                                 000100
    COLLECT STATISTICS FOR VEHICLE FINISHED WITH DEBOARDING                     000200
    FOR                                                                         000300
      EACH VEHICLE V" IN THE TRAIN OF V                                         000400
    DO                                                                          000500
      FOR                                                                       000600
        EACH TRIP(T) IN DEBOARD AND OUT (VDBLTL(V")) LIST                       000700
      DO                                                                        000800
        DQ T FROM VDBLTL(V")              <REMOVE TRIP FROM DEBOARD AND         000900
                                           LEAVE SYSTEM LIST>                   001000
        COLLECT STATISTICS FOR NUMBER OF TRIPS DEBOARDNG TO LEAVE               001100
        COLLECT STATISTICS FOR NUMBER OF PASSENGERS DEBOARDING TO LEAVE         001200
        INCREASE OCCUPANCY OF TL 3                                              001300
        UPDATE STATS FOR ENTERING ON TL 3                                       001400
        SET EVENT=3,TIME(TO INDICATE TO SAPFEL2 TO SKIP CERTAIN CODE),          001500
          & LINK=3                                                              001600
        SCHEDULE DEBOARD EXIT WALK                                              001700
      ENDDO                                                                     001800
      FOR                                                                       001900
        EACH TRIP(T) IN DEBOARD AND TRANSFER (VDXLTL(V")) LIST                  002000
      DO                                                                        002100
        DQ T FROM VDXLTL(V")              <REMOVE TRIP FROM DEBOARD AND         002200
                                           TRANSFER LIST>                       002300
        NQ T INTO BOARDING LIST           <PUT TRANSFER TRIP INTO BOARDING      002400
                                           MEMBERSHIP LIST>                     002500
        UOCC(3):=UOCC(3)+TPASS(T)         <INCREASE OCCUPANCY OF BOARDING       002600
                                           LINK BY SIZE OF TRIP>                002700
        COLLECT STATISTICS FOR NUMBER OF TRIPS DEBOARDING TO TRANSFER           002800
        COLLECT STATISTICS FOR NUMBER OF PASS DEBOARDING TO TRANSFER            002900
        COLLECT STATISTICS FOR A TRIP JOINING THE TRIP LINK                     003000
        SET EVENT=4,TIME(TO INDICATE TO SAPFEL2 TO SKIP CERTAIN CODE),          003100
          & LINK=3                                                              003200
        SCHEDULE TRANSFER WALK                                                  003300
      ENDDO                                                                     003400
    ENDDO                                                                       003500
                                                                                003600
```

```
      PROJECT: AGT                    MEMBER: SSMODA2                    DATE:
      LIBRARY: DSM                    LEVEL:  01.02                      TIME:
      TYPE:    PLL                    USERID: P326507                    PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   3    CASE(BOARD)
   5      COLLECT STATISTICS FOR A VEHICLE FINISHED WITH BOARDING
   6      DO VEHICLE FROM THE QUEUE OF BOARDING VEHICLES/TRAINS
   5      FOR
   7        EACH VEHICLE V" IN THE TRAIN OF V
   8      DO
   7        FOR
   9          EACH TRIP(T) IN BOARD (VBLTL(V")) LIST
   7        DO
   9          DO T FROM VBLTL(V")              <REMOVE TRIP FROM BOARD LIST>
   9          NC T INTO VTRIPQ(V)              <PUT TRIP INTO TRIP QUEUE ON
  41            _                               VEHICLE>
   9          DO T FROM UMEMTL(3)              <REMOVE TRIP FROM BOARD LINK>
   9          UOCC(3):=UOCC(3)-TPASS(T)        <DECREASE OCCUPANCY OF BOARDING
  41                                            LINK BY SIZE OF TRIP>
   9          COLLECT STATISTICS FOR THE NUMBER OF TRIPS BOARDING
   9          COLLECT STATISTICS FOR THE NUMBER OF PASSENGERS BOARDING
   9          COLLECT STATISTICS FOR A TRIP LEAVING THE TRIP LINK
   9          COLLECT STATISTICS FOR A TRIP LEAVING THE QUEUED STATE
   9          TGREAS (T):= 0
   9          VRES(V"):=0                      <MARK VEHICLE AS UNRESERVED>
   7        ENDDO
   5      ENDDO
   5      RUN SMMST(V)                         <FIND THE NEXT STOP OF THE HEAD
  41                                            VEHICLE IN THE TRAIN>
   5      RUN SUPMAC(3)                       <TRY TO GET TRIPS ON TURNSTILE
  41                                            TL MOVING INTO BOARDING TL.
  41                                            SINCE TRIPS HAVE LEFT BOARDING
  41                                            QUEUE TO BOARD V>
```

```
    PROJECT: AGT                    MEMBER: SSMODA3
    LIBRARY: DSM                    LEVEL:  01.06
    TYPE:    PDL                    USERID: C120496
ART
OL  ----+----1----+----2----+----3----+----4----+----5----+--

      CASE(JOINT)                                                                  000100
        COLLECT STATISTICS FOR VEHICLE FINISHED WITH BOARDING                      000200
        DQUE VEHICLE FROM THE QUEUE OF BOARDING VEHICLES                           000300
      5 FOR                                                                        000400
      7   EACH VEHICLE V" IN THE TRAIN OF V                                        000500
      5 DO                                                                         000500
      7   FOR                                                                      000700
      9     EACH TRIP(T) IN DEBOARD AND OUT (VDBLTL(V")) LIST                      000800
      7   DO                                                                       000900
      9     DQ T FROM VDBLTL(V")              <REMOVE TRIP FROM DEBOARD AND      001000
     11                                        LEAVE SYSTEM LIST>                 001100
      9     COLLECT STATISTICS FOR THE NUMBER OF TRIPS DEBOARDING TO             001200
     11       LEAVING THE STATION                                                 001300
      9     COLLECT STATISTICS FOR THE NUMBER OF PASSENGERS DEBOARDING TO        001400
     11       LEAVING THE STATION                                                 001500
      9     INCREASE OCCUPANCY OF TL 3                                           001600
      9     UPDATE STATS FOR ENTERING ON TL 3                                    001700
      9     SET EVENT=3,TIME(TO INDICATE TO SAFFEL2 TO SKIP CERTAIN CODE),       001800
     11       & LINK=3                                                          001900
      9     SCHEDULE DEBOARD EXIT WALK                                           002000
      7   ENDDO                                                                  002100
      7   FOR                                                                    002200
      9     EACH TRIP(T) IN DEBOARD AND TRANSFER (VDXLTL(V")) LIST               002300
      7   DO                                                                     002400
      9     DQ T FROM VDXLTL(V")              <REMOVE TRIP FROM DEBOARD AND      002500
     11                                        TRANSFER LIST>                     002600
      9     NQ T INTO BOARDING LIST           <PUT TRANSFER TRIP INTO BOARDING   002700
     11                                        MEMBERSHIP LIST>                   002800
      9     UOCC(3):=UOCC(3)+TPASS(T)         <INCREASE OCCUPANCY OF BOARDING    002900
     11                                        LINK BY SIZE OF TRIP>              003000
      9     COLLECT STATISTICS FOR THE NUMBER OF TRIPS DEBOARDING AND            003100
     11       TRANSFERRING                                                        003200
      9     COLLECT STATISTICS FOR THE NUMBER OF PASSENGERS DEBOARDING           003300
     11       AND TRANSFERRING                                                    003400
      9     COLLECT STATISTICS FOR A TRIP JOINING THE TRIP LINK                  003500
      9     SET EVENT=4,TIME(TO INDICATE TO SAFFEL2 TO SKIP CERTAIN CODE),       003600
     11       & LINK=3                                                          003700
      9     SCHEDULE TRANSFER WALK                                               003800
      7   ENDDO                                                                  003900
      7   FOR                                                                    004000
      9     EACH TRIP(T) IN BOARD (VBLTL(V")) LIST                               004100
      7   DO                                                                     004200
      9     DQ T FROM VBLTL(V")               <REMOVE TRIP FROM BOARD LIST>      004300
      9     NQ T INTO VTRIPQ(V)               <PUT TRIP INTO TRIP QUEUE ON       004400
     11                                        VEHICLE>                           004500
      9     DQ T FROM UMEMTL(3)               <REMOVE TRIP FROM BOARD LINK>      004600
      9     UOCC(3):=UOCC(3)-TPASS(T)         <DECREASE OCCUPANCY OF BOARDING    004700
     11                                        LINK BY SIZE OF TRIP>              004800
      9     COLLECT STATISTICS FOR THE NUMBER OF TRIPS BOARDING                  004900
      9     COLLECT STATISTICS FOR THE NUMBER OF PASSENGERS BOARDING             005000
      9     COLLECT STATISTICS FOR A TRIP LEAVING THE TRIP LINK                  005100
      9     COLLECT STATISTICS FOR A TRIP LEAVING THE QUEUED STATE               005200
      9     TQREAS (T):= 0                    <VEHICLE NOLONGER QUEUED>          005300
      9     VRES(V"):=0                       <MARK VEHICLE AS UNRESERVED>       005400
      7   ENDDO                                                                  005500
      5 ENDDO                                                                    005600
      5 RUN SMNXST(V)                         <FIND THE NEXT STOP OF THE HEAD    005700
     11                                        VEHICLE IN THE TRAIN>              005800
      5 RUN SUPMAC(3)                        <TRY TO GET TRIPS ON TURNSTILE      005900
     11                                        TL MOVING INTO BOARDING TL,        006000
     11                                        SINCE TRIPS HAVE LEFT BOARDING     006100
     11                                        QUEUE TO BOARD V>                  006200
                                                                                   006300
                                                                                   006400
```

```
     PROJECT: AGT                    MEMBER: SSMODB                    DATE:
     LIBRARY: LSM                    LEVEL:  01.36                     TIME:
     TYPE:    PLL                    USERID: PS26107                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC:SSMODB(V)
   3    <WHAT TO DO BEFORE TIME SEGMENT FOR THESE PROCESSES>
   1  ------------------------------------------------------------------------
   1  LOCAL VAR NAME  DIM     DESCRIPTION
   1  --------------- ------ ------------------------------------------------
   1  B1              -       BOARDING DELAY BASED ON NUMBER OF TRIPS BOARDING
   1  B2              -       BOARDING DELAY BASED ON THE SCHEDULED DEPARTURE
  26                           TIME
   1  B3              -       BOARDING DELAY BASED ON THE FORWARD VEHICLE'S
  26                           BOARDING DURATION
   1  D1              -       DEBOARDING DELAY BASED ON NUMBER OF TRIPS
  26                           DEBOARDING
   1  DESMAX          -       MAX DEBOARD/BOARD TIME
   1  DBTOTP          KMTLEN  THE NUMBER OF PASSENGERS DEBOARDING THE VEHICLE
   1  ORTTIM          -       OUTPUT RAMP TRAVEL TIME
   1  T               -       USED TO SPECIFY THE TIME THAT THE VEHICLE WILL
  25                           SPEND ON THE FEL
   1  T1              -       DELAY DUE TO MERGES ON THE REST OF
  26                           THE NETWORK
   1  T2              -       DELAY DUE TO LOCAL MERGES
   1  TLMAX           -       HEADWAY REQUIRED ON THE BYPASS LINK FOR THE
  26                           LONGEST POSSIBLE TRAIN
   1  TMAX            -       THE AMOUNT OF TIME A VEHICLE WILL SPEND ON THE
  26                           FEL FOR THE BOARD/DEBOARD/JOINT EVENT
   1  TMEAN           -       A MEAN TIME
   1  Q               -       AN INDICATOR WITH A VALUE OF 1.0 WHEN THE
  26                           CURRENT STATION LINK HAS A HEADWAY EVENT
  26                           AND A VALUE OF 0.0 WHEN IT HAS NO STATION LINK.
  26                           USED TO MAKE TRAVEL TIME A FUNCTION OF HEADWAY
  26                           TIME.
   1  VEFFOR          -       THE VEHICLE/LEAD VEHICLE IN THE TRAIN IN FRONT
  26                           OF THIS VEHICLE ON THE STATION LINK MEMBERSHIP
  26                           LIST
   1  ------------------------------------------------------------------------
   3    VGREAS(V):=0             <MARK V AS ABOUT TO BE PUT ON THE FEL>
   3    CASEENTRY(VMEVNT(V))     <BASED ON THE EVENT THAT IS ABOUT TO HAPPEN
  28                             TO THE VEHICLE>

   3    CASE(HEADWAY ZONE)
   5      SCHZF(SL):=T           <INDICATE HEADWAY ZONE CANNOT BE ENTERED>
   3      T:=SLMIA(SL)*VTRLEN(V)+SLHTB(SL) <CALC HEADWAY ZONE TRAVEL TIME>
   5      SCHED(V,T)             <SCHEDULE VEHICLE V TO SPEND TIME T ON FEL>

   3    CASE(TRAVEL)
   5      INCLUDE SSMODB1

   3    CASE(DEBOARD)
   5      INCLUDE SSMODB2

   3    CASE(BOARD)
```

```
      PROJECT: ACT                    MEMBER: SSMOD3                    DATE:
      LIBRARY: DSM                    LEVEL:  01.36                     TIME:
      TYPE:    PDL                    USERID: P326507                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--

    5       INCLUDE SSMOD83

    3     CASE(JOIN)
    5       INCLUDE SSMOD34

    3     CASE(STORE)
    5       VCREAS(V):=4              <INDICATES VEHICLE QUEUED IN STORAGE>

    3     CASE(LAUNCH)
    5       INCLUDE SSMOD85

    3     ENDCASE
    1   ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SSMODBT                    DATE:
      LIBRARY: USM                    LEVEL:  01.03                      TIME:
      TYPE:    PLL                    USERID: P326507                    PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    3     CASE (TRAVEL)
    3       IF
    7         FIRST EVENT ON THE VEHICLE'S CURRENT STATION LINK IS THE
    7         RAILWAY TRAVEL EVENT
    5       THEN
    7         T:= MAX(0.,SLPENT(SL)*(SLTTIM(SL)-SLHTA(SL)*VTRLEN(V)-SLHTB(SL)))
    5       ELSE
    7         T:= MAX(0.,SLPENT(SL)*SLTTIM(SL))
    5       ENDIF
    5       SCHED (V.T)
    5       IF
    7         PL2IND = T <LOCAL MERGE DELAY IS IN AFFECT> AND
    7         VSL(V) = 12 <THE VEHICLE IS ON THE BYPASS LINK>
    5       THEN
    7         IF
    9           THIS IS THE FIRST VEHICLE USING THE GAP LIST
    7         THEN
    9           DEFINE THE START OF THE FIRST GAP
    7         ELSE
    9           DEFINE THE END OF ONE GAP
    9           DECIDE WHETHER THAT GAP IS TOO SHORT TO HOLD A TRAIN
    9           IF
   11             THE GAP SIZE IS LESS THAN TLMAX
    9           THEN
   11             WRITE OVER THE LAST GAP
    9           ELSE
   11             CREATE A NEW GAP
   11             INCREMENT POINTER TO LAST GAP ADDED
    9           ENDIF
    9           GAP START = TAIL OF THIS TRAIN
    9           GAP END = 0
    7         ENDIF
    5       ENDIF
```

```
PROJECT: AGT                    MEMBER: SSMOD32
LIBRARY: DSM                    LEVEL:  01.09
TYPE:    PDL                    USERID: C120496
ART
.OL  ----+----1----+----2----+----3----+----4----+----5----+--

       CASE (DEBOARD)                                                          000100
         COLLECT STATS FOR A VEHICLE/EACH VEHICLE IN THE TRAIN, PENDING        000200
       5 DEBOARDING.                                                           000300
       5 IF                                                                    000400
       7   PENTS = T                                                           000500
       7   AND VTRNCH(V) NE 0  <V IS NOT A SOLITARY VEHICLE>                   000600
       7   AND VTRNCH(V) NE V                                                  000700
       5 THEN                                                                  000800
       7   RUN SMDETR (V)                                                      000900
       5 ELSE                                                                  001000
       7   ADLST (1) = 0                                                       001100
       5 ENDIF                                                                 001200
       5 I = 0                                                                 001300
       5 FOR V                                                                 001400
       7   THE FIRST VEHICLE/TRAIN AND EACH SUBSEQUENT VEHICLE IN ADLST        001500
       5 DO                                                                    001600
       7   TMAX = 0                                                            001700
       7   VID = V                                                             001800
       7   FOR VID                                                             001900
       9     EACH VEHICLE IN THE TRAIN OR FOR A SOLITARY VEHICLE               002000
       7   DO                                                                  002100
       9     RUN SMDBRD (VID)                                                  002200
       9     VNPASS (VID) = VNPASS (VID) - VTOTP (VID)                         002300
       9     TMEAN:= STDBA * VTOTP (VID) + STDBC                               002400
       9     VRANDN (AKSEED, TMEAN, STDBSD,D1)                                 002500
       9     TMAX = MAX (TMAX, D1)                                             002600
       9     VID = VTRNCH (VID)                                                002700
       7   UNTIL                                                               002800
       9     VID = V OR                                                        002900
       9     VID = 0                                                           003000
       7   ENDDO                                                               003100
       7   SCHED (V, TMAX)                                                     003200
       7   IF(WRITING TRIP & VEHICLE EVENT FILE)                               003300
       9     FOR                                                               003400
               EACH VEHICLE IN THE TRAIN                                       003500
             DO                                                                003600
      11       FOR                                                            003700
      13         EACH TRIP DEBOARDING TO LEAVE                                003800
      11       DO                                                             003900
      13         WRITE TVF RCD                                                004000
      11       ENDDO                                                          004100
      11       FOR                                                            004200
      13         EACH TRIP DEBOARDING TO TRANSFER                             004300
      11       DO                                                             004400
      13         WRITE TVF RCD                                                004500
      11       ENDDO                                                          004600
       9     ENDDO                                                             004700
       7   ENDIF                                                               004800
       7   I = I + 1                                                           004900
       7   V = ADLST (I)                                                       005000
       5 UNTIL                                                                 005100
       7   V = 0                                                               005200
       5 ENDDO                                                                 005300
```

```
   PROJECT: AGT                    MEMBER: SSMODB3
   LIBRARY: DSM                    LEVEL:  01.08
   TYPE:    PDL                    USERID: C120496
SRT
OL ----+----1----+----2----+----3----+----4----+----5----+----
   CASE   (BOARD)                                                    000100C
       COLLECT STATS FOR A VEHICLE /ALL VEHICLES IN THE TRAIN, PENDING 000200C
     5 BOARDING.                                                     000300C
     5 NQUE THE VEHICLE OR LEAD VEHICLE OF THE TRAIN IN THE LIST OF  000400C
     5 BOARDING VEHICLES/TRAINS.                                     000500C
     5 TMAX = 0                                                      000600C
     5 RUN SMBRD (V)                                                 000700C
                                                                     000800C
     5 VID = V                                                       000900C
     5 FOR VID                                                       001000C
     7   EACH VEHICLE IN THE TRAIN OR FOR A SOLITARY VEHICLE         001100C
     5 DO                                                            001200C
     7   VNPASS (VID) = VNPASS (VID) + VTOTP (VID)                   001300C
     7   TMEAN:= STBA*VTOTP (VID) + STBC                             001400C
     7   VRANDN (AKSEED, TMEAN, STBSD, B1)                           001500C
     7   TMAX = MAX (TMAX, B1)                                       001600C
     7   VID = VTRNCH (VID)                                          001700C
     5 UNTIL                                                         001800C
     7   VID = V OR                                                  001900C
     7   VID = 0                                                     002000C
     5 ENDDO                                                         002100C
     5 IF SERVICE POLICY IS SCHEDULED                                002200C
     7   POLSER = 3                                                  002300C
     5 THEN                                                          002400C
     7   RUN SMLTIM (V)                                              002500C
     7   TMAX:= MAX (TMAX, B2)                                       002600C
     7   VBEFOR:= THE VEHICLE IN FRONT OF V                          002700C
     7   IF                                                          002800C
     9     VBEFOR IS IN THE BOARD EVENT ON THE FEL                   002900C
     7   THEN                                                        003000C
     9     B3 = VTIME (VBEFOR) / 10 - CLOCK                          003100C
     9     TMAX:= MAX (TMAX,B3)                                      003200C
     7   ENDIF                                                       003300C
     7   PLSTLV(VROUTE(V)) = CLOCK + TMAX                            003400C
     5 ENDIF                                                         003500C
       SCHED (V, TMAX)                                               003600C
       IF(WRITING TRIP & VEHICLE EVENT FILE)                         003700C
     7   FOR                                                         003800C
     9     EACH VEHICLE IN THE TRAIN                                 003900C
     7   DO                                                          004000C
     9     FOR                                                       004100C
    11       EACH TRIP BOARDING                                      004200C
     9     DO                                                        004300C
    11       WRITE TVF RCD                                           004400C
     9     ENDDO                                                     004500C
     7   ENDDO                                                       004600C
     5 ENDIF                                                         004700C
```

```
PROJECT: AGT                    MEMBER: SSMODB4
LIBRARY: DSM                    LEVEL:  01.11
TYPE:    PDL                    USERID: C120496
ART
OL  ----+----1----+----2----+----3----+----4----+----5----+----6-

    CASE   (JOINT)                                                                  0001000
        COLLECT STATS FOR A VEHICLE /ALL VEHICLES IN THE TRAIN, PENDING             0002000
        BOARDING.                                                                   0003000
        IF                                                                          0004000
          PENTS = T AND                                                             0005000
          VTRNCH(V) ¬= 0 AND V                                                      0006000
        THEN                                                                        0007000
          RUN SMDETR (V)                                                            0008000
        ELSE                                                                        0009000
          ADLST (1) = 0                                                             0010000
        ENDIF                                                                       0011000
        J = 0                                                                       0012000
        FOR V                                                                       0013000
          THE FIRST VEHICLE/TRAIN AND EACH SUBSEQUENT VEHICLE IN ADLST              0014000
        DO                                                                          0015000
          NQUE EACH VEHICLE IN THE QUEUE OF BOARDING VEHICLES USING VBEVCH          0016000
          AND HEADED BY SBEQTL                                                      0017000
          I = 0                                                                     0018000
          TMAX = 0                                                                  0019000
          VID = V                                                                   0020000
          FOR VID                                                                   0021000
            EACH VEHICLE IN THE TRAIN OR FOR A SOLITARY VEHICLE                     0022000
          DO                                                                        0023000
            I = I + 1                                                               0024000
            RUN SMDBRD (VID)                                                        0025000
            VNPASS (VID) = VNPASS (VID) - VTOTP (VID)                               0026000
            DBTOTP(I) = VTOTP (VID)                                                 0027000
            VID = VTRNCH(VID)                                                       0028000
          UNTIL                                                                     0029000
            VID = V OR                                                              0030000
            VID = 0 OR                                                              0031000
          ENDDO                                                                     0032000
          RUN SMBRD (V)                                                             0033000
          I:= 0                                                                     0034000
          FOR VID                                                                   0035000
            EACH VEHICLE IN THE TRAIN OR FOR A SOLITARY VEHICLE                     0036000
          DO                                                                        0037000
            I = I + 1                                                               0038000
            VNPASS (VID) = VNPASS (VID) + VTOTP (VID)                               0039000
            TMEAN:= STDBA * DBTOTP(I) + FLDI4(1)*DBTOTP(I)*VTOTP(VID)               0040000
                    + STDBB*VTOTP(VID) + STDBC                                      0041000
            VRANDN (AKSEED, TMEAN, STDBSD, D1)                                      0042000
            TMEAN:= STBA * VTOTP(VID)+FLDI4(2)*DBTOTP(I)*VTOTP(VID)                 0043000
                    + STBB*DBTOTP(I)   + STBC                                       0044000
            VRANDN (AKSEED, TMEAN, STBSD, B1)                                       0045000
            DBBMAX = MAX (D1, B1 + STDLAY)                                          0046000
            TMAX = MAX (TMAX, DBBMAX)                                               0047000
            VID = VTRNCH (VID)                                                      0048000
          UNTIL                                                                     0049000
            VID = V OR                                                              0050000
            VID = 0                                                                 0051000
          ENDDO                                                                     0052000
          IF SERVICE POLICY IS SCHEDULED                                            0053000
            POLSER = 3                                                              0054000
          THEN                                                                      0055000
            RUN SMLTIM (V)                                                          0056000
            TMAX:= MAX (TMAX, B2)                                                   0057000
            VBEFOR:= THE VEHICLE IN FRONT OF V                                      0058000
            IF                                                                      0059000
              VBEFOR IS IN THE JOINT EVENT ON THE FEL                               0060000
            THEN                                                                    0061000
              B3 = VTIME (VBEFOR) / 10 - CLOCK                                      0062000
              TMAX:= MAX (TMAX,B3)                                                  0063000
            ENDIF                                                                   0064000
            PLSTLV(VROUTE(V)) = CLOCK + TMAX                                        0065000
          ENDIF                                                                     0066000
          SCHED (V, TMAX)                                                           0067000
          IF(WRITING TRIP & VEHICLE EVENT FILE)                                     0068001
            FOR                                                                     0069001
              EACH VEHICLE IN THE TRAIN                                             0070001
```

```
    PROJECT: AGT                    MEMBER: SSMODB4
    LIBRARY: DSM                    LEVEL:  01.11
    TYPE:    PDL                    USERID: C120496
ART
OL  ----+----1----+----2----+----3----+----4----+----5----+----6-

          DO                                                          007100
            FOR                                                       007200
15            EACH TRIP DEBOARDING TO LEAVE                           007300
11          DO                                                        007400
13            WRITE TVF RCD                                         007500
11          ENDDO                                                     007600
11          FOR                                                       007700
13            EACH TRIP DEBOARDING TO TRANSFER                        007800
11          DO                                                        007900
13            WRITE TVF RCD                                         008000
11          ENDDO                                                     008100
11          FOR                                                       008200
13            EACH TRIP BOARDING                                      008300
11          DO                                                        008400
13            WRITE TVF RCD                                         008500
11          ENDDO                                                     008600
 9        ENDDO                                                       008700
 7      ENDIF                                                         008800
 7      J = J + 1                                                     008900
 7      V = ADLST (J)                                                 009000
 5    UNTIL                                                          009100
 7      V = 0                                                         009200
 5    ENDDO                                                          009300
```

```
PROJECT: AGT                    MEMBER: SSMODB5                    DATE
LIBRARY: LSM                    LEVEL:  01.02                      TIME
TYPE:    PLI                    USERID: P326507                    PAGE:
START
COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   5     CASE(LAUNCH)
   6       IF  <THIS IS THE FIRST LAUNCH ATTEMPT>
   7         VLAGAN(V) = F
   8       THEN
   7         COLLECT STATISTICS FOR A VEHICLE PENDING ITS FIRST LAUNCH ATTEMPT
   7         DETERMINE LAUNCH DELAY DUE TO MERGES IN REST OF NET
   7         T1 = SMPSEL(PNMODT)
   7         SCHED (V, T1)
   7         VLAGAN(V) = T
   8       ELSE
   7         IF <THE LOCAL MERGE POLICY IS IN EFFECT>
   9           BL2IND = T
   7         THEN
   9           <FIND DELAY UNTIL A SLOT IN THE BYPASS LINK CAN BE FOUND, T2.>
   9           <IF A SLOT CAN'T BE FOUND, DETERMINE THE DELAY UNTIL RETRY
   9           IS ATTEMPTED, T2 AND SET VLAGAN(V) TO TRUE. IF RETRY IS SUCCESS
   9           FUL, SET VLAGAN(V) TO FALSE.>
   9           IF
  11             THE LEAD VEHICLE ON THE BYPASS LINK IS QUEUED
   9           THEN
  11             T2 = STTIM (BYPASS) - STTIM (OUTPUT RAMP)
  11             VLAGAN:= T
   9           ELSE
  11             WHILE
  13               THE GAP END IS LESS THAN THE VEHICLE'S ASSUMED GAP END IF
  13               IT WERE NOW LAUNCHED, AND
  13               THE LAST GAP IN THE TABLE IS NOT BEING CHECKED
  11             DO
  13               INCREMENT THE POINTER TO THE LAST AVAILABLE GAP
  11             ENDDO
  11             IF
  13               THE GAP BEING CHECKED IS THE LAST AVAILABLE GAP IN THE TABL
  11             THEN
  13               ONLY ALLOW LAUNCH TO OCCUR IF IT CAN BE GUARANTEED THAT NO
  13               TRAIN WILL COME OFF THE FEL ON THE BYPASS LINK BEFORE THE
  13               TAIL OF THE LAUNCHED VEHICLE
  13               IF
  15                 CLOCK + MIN TRAVEL TIME >= THE POTENTIAL NEW GAP START
  15                 + MAX TRN LENGTH...OR EQUIVALENTLY...CLOCK + STTIM - TLMA
  15                 >= GAP START + TLMAX
  13               THEN
  15                 ALLOW THE LAUNCH AND UPDATE THE GAP TABLE
  15                 VLAGAN (V) := F
  15                 IF
  17                   SLOT START TIME > CLOCK + OUTPUT RAMP TRAVEL TIME
  17                   T2 = SLOT START TIME - CLOCK - OUTPUT RAMP TRAVEL TIME
  15                 ELSE
  17                   T2 = 0
  15                 ENDIF
  15                 SET THE START OF THE LAST AVAILABLE GAP TO THE GAP CREATE
```

```
      PROJECT: ACT                    MEMBER: SSMCDB5                    DATE:
      LIBRARY: DSM                    LEVEL:  01.02                      TIME:
      TYPE:    PSL                    USERID: P326507                    PAGE:
START ,
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  15                  BY THIS VEHICLE:
  15                  CLOCK + ORTTIM + TLMAX + T2
  13                ELSE
  15                  VLAGAN (V) := 1
  15                  T2 = GAP START OF THE LAST AVAILABLE GAP - CLOCK - ORTTIM
  13                ENDIF
  11              ELSE
  13                VLAGAN (V):= F
  13                IF
  15                  SLOT START TIME > CLOCK + ORTTIM
  13                THEN
  15                  T2 = SLOT START TIME - CLOCK - ORTTIM
  13                ELSE
  15                  T2 = 0
  13                ENDIF
  13                NEW SLOT START TIME = CLOCK + ORTTIM + T2 + TLMAX
  13                <THE TIME REQUIRED FOR THE TAIL OF THE LAUNCHED VEHICLE TO
  13                COME OFF THE OUTPUT RAMP>
  13                IF
  15                  THE NEW SLOT TIME < TLMAX
  13                THEN
  15                  INCREMENT THE NEXT AVAILABLE GAP POINTER
  13                ELSE
  13                ENDIF
  11              ENDIF
   9            ENDIF
   9            SCHED (V, T2)
   7          ENDIF
   5        ENDIF
```

```
      PROJECT: AGT                          MEMBER: SSMODN                          DATE
      LIBRARY: GSM                          LEVEL:  01.27                           TIME
      TYPE:    PDL                          USERID: P326507                         PAGE
STANT
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  ------------------------------------------------------------------------
    1  PROC:SSMODN
    3    <DETERMINE NEXT EVENT FOR V OR SIGNAL 'DONE'>
    1  ------------------------------------------------------------------------
    1  LOCAL VAR NAME  DIM     DESCRIPTION
    1  --------------- ------- ------------------------------------------------
    1  EVFND           -       USED TO INDICATE:
   26                             F===>SKIP OVER DEBOARD AND BOARD EVENTS IN
   33                                    THE CASE OF AN ON-LINE STATION WHERE
   33                                    THE VEHICLE IS NOT TO STOP
   26                             T===>DO NOT SKIP OVER DEBOARD & BOARD EVENTS
    1  ------------------------------------------------------------------------
    3    IF   <FIRST EVENT ON THIS SL FOR V>
    5      V.EVNT(V)=0
    3    THEN
    5      NO V INTO SLMEMT(SL)          <PUT V ON MEMBERSHIP CHAIN OF SL>
    5      SLOCC(SL):=SLOCC(SL)+VTRLEN(V) <INCREASE OCCUPANCY
   53                                                    OF SL BY TRAIN LEN>
    5      COLLECT STATISTICS FOR A TRAIN ENTERING THE STATION LINK.
    5      IF  <THE CURRENT SL OF V HAS DEBOARD AND/OR BOARD EVENTS>
    7        SLTYPE (SL) = 3
    5      THEN
    7        SLPOCC(SL):=SLPOCC(SL)+VTRLEN(V) <INCREASE PSEUDO-OCCUPANCY OF SL
    7        BY TRAIN  LENGTH OF V>
    5      ENDIF
    3    ENDIF

    3    EVFND:=F          <SET 'EVFND' TO SKIP OVER DEBOARD & BOARD EVENTS>
    3    DO
    5      V.EVNT(V):=NEXT SLEVL(SL)

    5      CASE=TRY(V.EVNT(V))       <BASED ON THE NEXT VEHICLE EVENT TO BE
   30                                PERFORMED>

    5      CASE(HEADWAY ZONE OR TRAVEL OR STORE)
    7        EVFND:=T      <TURN SKIP FLAG OFF, SO THIS EVENT IS PERFORMED>

    5      CASE(DEBOARD, BOARD, OR JOINT)
    7        IF
    9          STYPE=F     OR        <STATION IS OFF-LINE>
    9          VNXSTN(V)=STSIM         <VEHICLE IS SUPPOSED TO STOP HERE>
    7        THEN
    9          EVFND:=T <TURN SKIP FLAG OFF, SO THIS EVENT IS PERFORMED>
    7        ELSE
    9          EVFND:=F                <SKIP THIS EVENT>
    7        ENDIF

    5      CASE (LAUNCH)
    7        IF      <V NOT AT HEAD OF SL>
    9          V.EMCH(SLMEMT(SL))¬=V
```

```
      PROJECT: AGT                        MEMBER: SSMODN                        DATE:
      LIBRARY: GSM                        LEVEL:  01.27                         TIME:
      TYPE:    PEL                        USERID: P326507                       PAGE:
START
 COL  ---+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   7          THEN
   9            VGREAS(V):=3          <MARK V QUEUED WAITING TO START LAUNCH
  29                                  EVENTS>
   7          ENDIF
   7          EVFND:=T      <TURN SKIP FLAG OFF, SO THIS EVENT IS PERFORMED>

   5        CASE(0)     <NO MORE EVENTS ON EVENT INDICATOR LIST FOR THIS SL>
   7          IF <V HAD BEEN WAITING TO START ITS LAUNCH EVENT(S)>
   9            VGREAS(V)=3 OR
   9            VLAGAN(V)=T     <COULDN'T FIND SLOT AND REQUIRES ANOTHER LAUNCH>
   7          THEN
   9            V.NEVNT(V):=LAUNCH
   7          ELSE <NO MORE EVENTS ON THIS LINK>
   9            ADONES:=T           <SIGNAL 'DONE'>
   7          ENDIF
   7          EVFND:=T      <TURN SKIP FLAG OFF, SO THIS EVENT IS PERFORMED>

   5        ENDCASE
   3      UNTIL
   5        EVFND = T
   3      ENDDO
   1    ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SSPMAC                    DATE
      LIBRARY: SSM                    LEVEL:  01.02                     TIME
      TYPE:    PLL                    USERID: C120496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+---- 7-

    1  PROC:SSPMAC(SL*,FLAG)
    2    DO PRELIMINARY PROMPT TESTING
    3    IF
    4      PROMPT IS FEASIBLE
    5    THEN
    6      GET A FREE XTN (X*) TO SCHED SLPROMPT(SL*,FLAG)
    7      SEVENT(X*):=SLPROMPT
    8      SCHED(X*,0)
    9    ENDIF
   10  ENDPROC
```

```
     PROJECT: AGT                    MEMBER: SSTEST                    DATE
     LIBRARY: GSM                    LEVEL:  01.24                     TIME:
     TYPE:    PEL                    USERID: P326507                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC:SSTEST(V)
   2    <FIND THE NEXT SL & TEST IF CAN ENTER (IF NOT, FIND ALTERNATE;ETC.)>

   3    SL:=VSL(V)
   4    AENTRS:=F  <ASSUME THAT THE VEHICLE CANNOT ENTER FURTHER LINKS UNTIL
   5    THE FOLLOWING APPROPRIATE TESTS HAVE BEEN PASSED>
   6    VGREAS(V):=0  <ASSUME THAT THE VEHICLE IS NOT QUEUED UNTIL PROVEN TO
   7    BE>

   8    IF   <V IS NOT AT TOP OF ITS SL>
   9      VMEMCH (SLMEMT (SL)) =: V
  10    THEN
  11      VGREAS(V):=2    <V Q'D DUE TO OTHER VEH IN FRONT & OTHERWISE DONE>
  12    ELSE   <V IS AT TOP OF ITS SL>
  13      IF   <EXIT OF V'S SL IS FAILED>
  14        SLEXIT(SL)=T
  15      THEN
  16        VGREAS(V):=1 <V Q'D DUE TO CONG.,FAILED EXIT, OR FAILED   ENTRY>
  17      ELSE   <EXIT OF V'S SL IS NOT FAILED>
  18        IF   <THERE IS NOT A DIVERGE>
  19          SLDIVC(SL)=0
  20        THEN
  21          ASLLST:=SL'S SINGLE DOWNSTREAM SL
  22        ELSE  <THERE IS A DIVERGE>
  23          ASLLST:=SL'S MULTIPLE DOWNSTREAM STATION LINKS AND SINKS
  24          STARTING WITH SLDSL(SLDSP(SL))
  25          RUN SMDIVF(V,SLDIVC(SL),ASLLST) <RETURN ASLLST ORDERED BY
  26          PREFERENCE AND ENDING IN ZERO.>
  27        ENDIF
  28        FOR
  29          EACH MEMBER, (I), OF ASLLST OF POSSIBLE LINKS
  30        DO
  31          IF   <NEXT SL IS A SINK>
  32            ASLLST(I)=SINK
  33          THEN
  34            AENTRS:=T               <SET TO CAN ENTER>
  35          ELSE   <NEXT SL IS NOT A SINK>
  36            IF
  37              SLAVAL(ASLLST(I))=T & <SL IS AVAILABLE>
  38              SLENT(ASLLST(I))=F     <SL ENTRY IS NOT FAILED>
  39            THEN
  40              IF <SL HEADWAY ZONE NOT OCCUPIED ON THIS SL>
  41                SLHZF(ASLLST(I))=F
  42              THEN
  43                IF <ASLLST(I) HAS DS/S EVENTS>
  44                  SLTYPE (ASLLST(I)) = 3
  45                THEN
  46                  IF <V'S TRAIN CAN FIT IN THESE BERTHS>
  47                    SLCAP(ASLLST(I))>=SLPOCC(ASLLST(I))+VTRLEN(V)
  48                  THEN
```

```
      PROJECT: AUT                    MEMBER: SSTEST                    DATE:
      LIBRARY: DSM                    LEVEL:  01.24                     TIME:
      TYPE:    PSL                    USERID: P326507                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  19                    AENTRS:=T          <SET TO CAN ENTER>
  17                  ENDIF
  15                ELSE <ASLLST(I) DOES NOT HAVE DB/B EVENTS>
  17                  IF <V'S TRAIN CAN FIT ON THIS DOWNSTREAM SL>
  19                    SLCAP(ASLLST(I))>=SLOCC(ASLLST(I))+VTMLEN(V)
  17                  THEN
  19                    AENTRS:=T          <SET TO CAN ENTER>
  17                  ENDIF
  15                ENDIF
  13              ENDIF
  11            ENDIF
   9          ENDIF
   7        UNTIL
   9          AENTRS=T OR  <THE VEHICLE CAN ENTER THE LINK OR
   9          ASLLST(I) = 0  <THE LAST STATION LINK ON THE LIST HAS BEEN
   9          CHECKED>
   7        ENDDO
   7        IF <THE VEHICLE COULD ENTER ONE OF THE DOWNSTREAM LINKS>
   9          AENTRS = T
   7        THEN
   9          IF
  11            THE VEHICLE IS ON THE APPROACH LINK  AND
  11            ASLLST (I) = INPUT RAMP
   9          THEN
  11            IF
  13              ASLLST (I) = BYPASS LINK
  11            THEN
  13              COLLECT STATION ENTRY REJECTION STATISTICS
  11            ELSE
  13              COLLECT STATION ENTRY ACCEPTANCE STATISTICS
  11            ENDIF
   9          ENDIF
   9          AVAXSL:=ASLLST(I) <THE NEXT STATION LINK TO BE ENTERED HAS BEEN
   9          DETERMINED>
   7        ELSE  <THE VEHICLE IS QUEUED DUE TO CONGESTION>
   9          VQUEAS(V):=1
   7        ENDIF
   5      ENDIF
   3    ENDIF
   1  ENDPROC
```

```
      PROJECT: AGT                        MEMBER: SULEAV                        DATE:
      LIBRARY: SUM                        LEVEL:  01.16                         TIME:
      TYPE:    PSL                        USERID: P326507                       PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-
    1  PROC:SULEAV(T) <PROCESS A TRIP LEAVING A TICKETING OR TURNSTILE
   17                 TRIP LINK>
    3    NEXTT:=UMENTL(T) <THE TRIP IMMEDIATELY BEHIND T ON THIS TL>
    3    UCCC(TCURR(T)):=UCCC(TCURR(T))-TPASS(T)
    3    OUT T FROM UMENTL(TCURR(T)) <MEMBERSHIP CHAIN>
    3    SUPMAC(TCURR(T))
    3    COLLECT STATISTICS FOR THE TRIP LEAVING THE TRIP LINK
    3    IF              <IF THERE IS A FOLLOWING TRIP>
    5      NEXTT =/= T
    3    THEN
    5      CASEENTRY(TGREAS(NEXTT))
    5      CASE(3) <NEXTT HAS BEEN WAITING TO START ITS PROCESSING EVENT>
    7        <NOTE: SEVENT(NEXTT) ALREADY SET TO TRIP EVENT &
   14                TEVENT(NEXTT) ALREADY SET TO 1>
    7        RUN SUMSD(NEXTT)
    5      CASEELSE
    7        <NO OP> <NOTE: CASE=0 ===> TNEXT ON FEL, BUT IT CANNOT BE SINCE
   35                                    IT WAS NOT AT HEAD OF TCURR(T)
   22                        CASE=1 ===> NEXTT QUEUED DUE TO CONG./FAIL, BUT IT
   34                                    CANNOT BE SINCE IT WAS NOT AT THE
   34                                    HEAD OF ITS TCURR(T)
   22                        CASE=2 ===> TNEXT QUEUED DUE TO TRIPS IN FRONT OF
   34                                    IT & OTHERWISE DONE, BUT IT CANNOT BE
   34                                    DONE SINCE IT WAS NOT AT THE HEAD
   34                                    OF ITS TL AND THUS COULD NOT START
   34                                    ITS PROCESSING EVENT
   22                        CASE=4 ===> NOT APPLICABLE
    5      ENDCASE
    3    ENDIF
    1  ENDPROC
```

```
    PROJECT: AGT                        MEMBER: SUMOD
    LIBRARY: DSM                        LEVEL:  01.14
    TYPE:    PLL                        USERID: C12049o
ART
OL  ----+----1----+----2----+----3----+----4----+----5----+----6-

    PROC: SUMOD(T)                                                                  000 10
    ------------------------------------------------------------------------------- 000 20
  . LOCAL VAR NAME  DIM    DESCRIPTION                                              000 30
  1 ------------------ ------ ------------------------------------------------------ 000 40
  1 TIM               -      TIME TO PERFORM THE PROCESSING EVENT                    000 50
  1 ------------------------------------------------------------------------------- 000 60
  3   TL:=TCURR(T)  <INITIALIZE TRIP'S CURRENT TRIP LINK TO "TL">                   000 70
  3   IF(EVENT IS DEBOARD EXIT WALK OR TRANSFER WALK)                               000 80
  5     IF(DESOARD EXIT WALK)                                                       000 90
  7       DECREASE THE OCCUPANCY OF THE TRIP LINK                                   001 00
  7       UPDATE STATS TO REFLECT TRIP LEAVING STN                                  001 10
  7       UPDATE STATS TO REFLECT TRIP LEAVING ON TL 3                              001 20
  7       UPDATE STATS TO REFLECT TRIP LEAVING FEL                                  001 30
  7       FREE XTN                                                                  001 40
  5     ELSE                                                                        001 50
  7       UPDATE STATS TO REFLECT TRIP LEAVING FEL                                  001 60
  7       RUN SMTABQ                                                                001 70
  3   ELSE                                                                          001 80
  5     ADONET:=F                                 <NOT DONE>                        001 90
  5     <THE TRIP LINK EVENT FOR WHICH THE TRIP HAS SPENT TIME ON THE FEL           002 00
  5     IS EVALUATED>                                                             002 10
  5     CASEENTRY(TMEVNT(T))                                                      002 20
  5     CASE(0)                                   <START WALK>                    002 30
  7       TMEVNT(T):=1                                        <SET NEXT TRIP EVENT> 002 40
  7       UOCC(TL):=UOCC(TL)+TPASS(T)    <INCREASE OCCUPANCY>                     002 50
  7       NQ T INTO UMEMTL(TL)               <MEMBERSHIP CHAIN>                   002 60
  7       TIM:=UTIM(TL)                                                           002 70
  7       SCHED(T,TIM)                                                            002 80
  5     CASE(1)                                   <END WALK, START PROCESS>       002 90
  7       IF <THE TRIP EVENT IS ON THE TICKETING LINK OR TURNSTILE LINK>          003 00
  9         TL = 1 OR TL = 2                                                      003 10
  7       THEN                                                                    003 20
  9         IF <SERVERS ARE AVAILABLE AND T IS AT THE HEAD OF THE TL>             003 30
 11           USERV(TL)>0 AND                                                    003 40
 1            THENCH(UMEMTL(TL))=T                                               003 50
            THEN                                                                003 60
 1            TIM = UTIMA(TL)*TPASS(T)/USERV(TL) + UTIMB(TL)                     003 70
30                                 <CALCULATE PROCESS TIME>                     003 80
 11           TOREAS(T):=0     <MARK AS ON FEL>                                  003 90
 11           TMEVNT(T):=2       <SET NEXT TRIP EVENT>                           004 00
 11           SCHED(T,TIM)       <SCHEDULE T TO SPEND TIME='TIM' ON FEL>         004 10
  9         ELSE <NO SERVERS AVAILABLE AND/OR NOT AT THE HEAD OF TL>              004 20
 11           TOREAS(T):=3                                                       004 30
  9         ENDIF                                                               004 40
  7       ELSE <TL=3> <THE TRIP'S LINK IS THE BOARDING LINK>                    004 50
 20                    <ALSO THE TRIP IS DONE WITH THE TRIP LINK MODEL>         004 60
  9         ADONET:=T                                                           004 70
  7       ENDIF                                                                 004 80
                                                                                004 90
  5     CASE(2) <THE TURNSTILE LINK OR TICKETING LINK IS DONE>                  005 00
  7       ADONET:=T                                                             005 10
                                                                                005 20
  5     ENDCASE                                                                 005 30
  3   ENDIF                                                                     005 40
  1 ENDPROC                                                                     005 50
```

```
     PROJECT: ACT                    MEMBER: SUBMAC                    DATE:
     LIBRARY: DSM                    LEVEL:  01.01                     TIME:
     TYPE:    PLL                    USERID: C120496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC:SUBMAC(TL*)
   3    DO PRELIMINARY PROMPT TESTING
   3    IF
   5      PROMPT FEASIBLE
   3    THEN
   5      GET A FREE XTN (X*) TO SCHED TLPROMPT(TL*)
   5      SEVENT(X*):=TLPROMPT
   5      SCHED(X*,0)
   3    END;IF
   1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SUTEST                    DATE
      LIBRARY: USM                    LEVEL:  01.14                     TIME
      TYPE:    PSL                    USERID: P326507                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  PROC:SUTEST(T)
    3    <NOTE:  THIS ROUTINE WOULD NEVER BE ENTERED FOR A TRIP THAT
   11             IS NOT AT THE HEAD OF ITS TRIP LINK>

    3    CASEENTRY(TCURR(T))  <DEPENDING ON THE TRIP LINK THAT THE TRIP
    3    IS CURRENTLY ON, DETERMINE WHETHER OR NOT IT CAN ENTER ITS NEXT
    3    TRIP LINK.  IF NOT, QUEUE IT ON ITS CURRENT LINK>

    3    CASE(0,1,2) <JUST ARRIVING AT STN,CURRENTLY IN TICKETING TL,
   16                  CURRENTLY IN TURNSTILE LINK>
    5      IF <CAN FIT>
    7        UCAP(TCURR(T)+1)>=UOCC(TCURR(T)+1)+TPASS(T)
    5      THEN
    7        AENTRT:=T                              <SET TO CAN ENTER>
    7        ATNXTL:=TCURR(T)+1                     <SET TRIP LINK TO NEXT TL>
    7        TGREAS (T):= 0
    5      ELSE
    7        AENTRT:=F                              <SET TO CANNOT ENTER>
    7        TGREAS(T):=1                          <MARK AS QUEUED DUE TO CONGEST>
    5      ENDIF

    3    CASE(3) <CURRENTLY IN BOARDING LINK>
    5      AENTRT:=T                                <SET TO CAN ENTER>
    5      ATNXTL:=4                                <MARK A AT THE END OF TL'S>
    5      TGREAS (T):= 0
    3    ENDCASE

    4  ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SZHDR                    DATE
      LIBRARY: DSM                    LEVEL:  01.00                    TIME
      TYPE:    PCL                    USERID: C120496                  PAGE
START
 COL  -+----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  PROC:SZHDR(NFULL,NTYPE)
    3    <WRITE HEADER RECORD TO RAW STATISTICS FILE SHOWING NUMBER OF
    4     FOLLOWERS AND TYPE OF HEADER>
    4     FORMAT RECORD
    4     WRITE RECORD
    1  ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SZINT                    DATE
      LIBRARY: LSM                    LEVEL:  01.00                    TIME
      TYPE:    PL1                    USERID: CI2049O                  PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  PROC:SZINT
    2    END-POINT CORRECT CURRENT TIME INTEGRALS(TIN)
    3    CALCULATE AVERAGES(ALIZANI)
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SZPLOT                  DATE:
      LIBRARY: JSM                    LEVEL:  00.01                   TIME:
      TYPE:    PCL                    USERID: H63954Z                 PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC ZPLOT <PRODUCE A TIME SERIES PLOT OF SAMPLED ITEMS>
    2    SITE=NIL LIMITS OF GRAPH & SCALING FACTOR
    3    RUN GRAPH <DISPLAY BIN CONTENTS>
    4  ENDPROC
```

```
    PROJECT: AGT                        MEMBER: SZREAD
    LIBRARY: DSM                        LEVEL:  00.02
    TYPE:    PDL                        USERID: C120496
ART
OL  ----+----1----+----2----+----3----+----4----+----5----+----6

    PROC ZREAD <BEGIN DATA ACQUISITION PROCESS>                                    0001000
         <READ DATA FROM RAW STATISTICS FILE USING PRE-ISSUED REQUESTS &           0002000
        START/STOP INTERVAL ENTERED VIA A READ COMMAND>                            0003000
                                                                                   0004000
3     IF <INITIAL ENTRY FOR DATA ACQUISITION>                                      0005000
5       ¬FILE                                                                      0006000
3     THEN <ACQUIRE SYSTEM CHARACTERISTICS>                                        0007000
5       READ HEADER                                                                0008000
5       IF <NOT SYSTEM HEADER>                                                     0009000
7         MTYPE¬=0                                                                 0010000
5       THEN <ERROR EXISTS IN FILE FORMAT>                                         0011000
7         ERROR - NO SYSTEM HEADER FOUND                                           0012000
5       ELSE <READ SIMULATION CHARACTERISTICS>                                     0013000
7         READ FOLLOWER <KNL,KNS,CSIZE,CSMAPL,KNV,KNR,SLTYPES&ZM220-ZM224>         0014000
7         CUSEC:=CSIZE/60.0  <SIMULATATOR TIME UNIT CONVERSION FACTOR>             0015000
7         FILE:=TRUE                                                               0016000
5       ENDIF                                                                      0017000
3     ELSE <DATA ACQUISITION INITIATED BY READ COMMAND>                            0018000
5       RUN SETUP <INITIALIZE MAJOR CATAGORY & SUB CATAGORY TABLES>                0019000
5       MSTART:=START*CUSEC    <CONVERT USER START/STOP TIME FROM MIN TO            0020000
5       MSTOP:=STOP*CUSEC           SIMULATION TIME UNITS>                         0021000
5       REWIND RAW STATISTICS FILE                                                 0022000
5       READ HEADER                                                                0023000
5       WHILE <REQUEST INTERVAL NOT FOUND>                                         0024000
7         MCLOCK<MSTART                                                            0025000
5       DO <SKIP TO BEGINNING OF REQUIRED INTERVAL>                                0026000
7         IF <FOLLOWER RECORDS EXIST>                                              0027000
9           NFOLL>0                                                                0028000
7         THEN <SKIP FOLLOWER RECORDS>                                             0029000
9           FOR <NUMBER OF FOLLOWERS>                                              0030000
11            REC=1 TO NFOLL                                                       0031000
9           DO                                                                     0032000
11            RUN SKIPFO                                                           0033000
9           ENDDO                                                                  0034000
7         ENDIF                                                                    0035000
7         RUN HEADER <READ NEXT HEADER>                                            0036000
5       ENDDO                                                                      0037000
5       <INITIALIZE MAJOR CATAGORY INDICATOR TABLE(MAJTBL)>                        0038000
5       WHILE <END OF INTERVAL NOT REACHED>                                        0039000
7         MCLOCK<=MSTOP                                                            0040000
5       DO <PERFORM DATA ACQUISITION & STORAGE>                                    0041000
7         IF <RECORD MAY BE NEEDED>                                                0042000
9           MAJTBL(MTYPE)¬=0                                                       0043000
7         THEN <IS RECORD REQUIRED>                                                0044000
9           IF <NEED NOT YET DETERMINED>                                           0045000
11            MAJTBL(MTYPE)=-1                                                     0046000
9           THEN                                                                   0047000
11            RUN REQTLU <DETERMINE IF RECORD IS REQUIRED>                         0048000
9           ENDIF                                                                  0049000
9           CASEENTRY-(MTYPE) <RECORD TYPE>                                        0050000
11            CASE (2) <SYSTEM RECORD>                                             0051000
13              RUN READ02                                                         0052000
11            CASE (2) <STATION LINK RECORD>                                       0053000
13              RUN READ03                                                         0054000
11            CASE (3) <TRIP LINK RECORD>                                          0055000
9           ENDCASE                                                                0056000
7         ENDIF                                                                    0057000
7         READ NEXT HEADER                                                         0058000
5       ENDIF                                                                      0059000
3     ENDDO                                                                        0060000
3     RUN RCLEAN <REINITIALIZE BIN STORAGE POINTERS>                               0061000
1   ENDPROC                                                                        0062000
```

```
      PROJECT: ACT                    MEMBER: SZREGTLU                    DATE
      LIBRARY: DSM                    LEVEL:  01.00                       TIME
      TYPE :   PCL                    USERID: P639547                     PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

   1  PROC REGTLU <DATA REQUEST/RECORD TYPE CORRELATION>
   6       <DETERMINE WHETHER A SPECIFIC RECORD TYPE CAN SATISFY A PARTIC-
   7        ULAR DATA REQUEST. THOSE REQUESTS WHICH CAN BE SATISFIED BY
   7        THE RECORD TYPE ARE CHAINED TOGETHER TO ALLOW FINDING THE
   7        REQUESTS IN SUBSEQUENT DATA RETRIEVALS WITHOUT REQUIRING A
   7        SEARCH OF THE REQUEST TABLE>

   3    PRIOR:=0
   3    FOR <ALL ENTRIES IN REQUEST TABLE>
   5      REG=1 TO REGNO
   3    DO <DETERMINE IF RECORD CAN SATISFY REQUEST>
   5      IF <MAIN CATAGORY MNEMONIC=CURRENT CATAGORY>
   7        REQTBL(REG,3)=MNTBL(MTYPE)
   5      THEN <SUBSTITUTE NUMERIC CATAGORY IDENTIFIER FOR MNEMONIC>
   7        REQTBL(REG,3)=MTYPE
   7        FOR <ALL SUBCATAGORIES POSSIBLE IN MAJOR CATAGORY>
   9          SUB=1 TO END OF TABLE
   7        DO <FIND SUBCATAGORY MNEMONIC & SUBSTITUTE NUMERIC ID>
   9          IF <MNEMONIC MATCH>
  11            REQTBL(REG,4)=SNTBL(SUB)
   9          THEN <SUBSTITUTE INDEX>
  11            REQTBL(REG,4):=SUB
  11            IF <PRIOR REQUEST NEEDING THIS DATA FOUND>
  13              PRIOR>0
  11            THEN <CHAIN REQUESTS>
  13              REQTBL(PRIOR,5):=REG
  11            ENDIF
  11            PRIOR:=REG
   9          ENDIF
   7        UNTIL
   9          SUBCATAGORY INDEX DETERMINED
   7        ENDDO
   5      ENDIF
   3    ENDDO
   1  PROCEND
```

```
      PROJECT: AGT                    MEMBER: SZSTAT                    DATE
      LIBRARY: DSM                    LEVEL:  01.11                     TIME
      TYPE:    PLL                    USERID: CI20490                   PAGE
START
 COL  ---+----1----+----2----+----3----+----4----+----5----+----6----+----7;

   1  PROC:SZSTAT    <COLLECT STATISTICS>

  10            I===>ELEMENT TYPE
  12              1===>VEHICLE
  12              2===>TRIP

  10            J===>ELEMENT NUMBER
  12              1<---><NV FOR VEHICLES
  12              1<---><NT FOR TRIPS

  10            K===>ENTITY TYPE
  12              1===>STATION (ENTIRE MODELLED AREA AS A WHOLE) (V&T)
  12              2===>STATION LINK  (V ONLY)
  12              3===>TRIP LINK     (T ONLY)

  10            L===>STATE DIRECTION
  12              1===>ENTERING STATE
  12              2===>LEAVING STATE

  10            M===>STATE
  12              FOR STATIONS
  14                1===>IN STATION
  14                2===>IN BOARD EVENT
  14                3===>IN DEBOARD EVENT
  14                4===>IN LAUNCH EVENT
  12              FOR STATION LINKS
  14                1===>ON STATION LINK
  14                2===>ON FEL
  14                3===>QUEUED
  12              FOR TRIP LINKS
  14                1===>ON TRIP LINK
  14                2===>ON FEL
  14                3===>QUEUED

  10            N===>LINK NUMBER
  12              0 FOR STATIONS
  12              1<---><MSL FOR STATION LINKS
  12              1<---><MTL FOR TRIP LINKS


   3    DIRECTION
   3    &CASEENTRY(L)

   3    ENTERING STATE
   3    &CASE(1)
   5      RUN SZSTATE

   3    LEAVING STATE
   3    &CASE(2)
   5      RUN SZSTATL
```

```
      PROJECT: ACT                    MEMBER: SZSTAT                  DATE:
      LIBRARY: LSM                    LEVEL:  01.11                   TIME:
      TYPE:    PLI                    USERID: C12049G                 PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  .     GEN.CASE
  1   ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SZSTATE                 DATE
      LIBRARY: DSM                    LEVEL:  01.07                   TIME
      TYPE:    PDL                    USERID: CI20496                 PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  PROC:SZSTATE
    3    SZSTATE - ENTERING STATE
    4      STATION/STATION LINK/TRIP LINK
    5      #CASEENTRY(X)

    6      STATION STATE
    6      #CASE(1)
    7        RUN SZSTATEN

    6      STATION LINK STATE
    6      #CASE(2)
    7        RUN SZSTATES

    6      TRIP LINK STATE
    6      #CASE(3)
    7        RUN SZSTATET
    6      #ENDCASE
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SZSTATEN                    DATE
      LIBRARY: GSM                    LEVEL:  01.07                       TIME
      TYPE:    PLC                    USERID: CL0496                      PAGE
STANT
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SZSTATEN
    2    SZSTATEN - ENTERING STATION STATE
    3    VEHICLE/TRIP
    4    CASEENTRY(I)

    5    VEHICLE
    6    CASE(1)
    7      ZNVNE(M)=ZNVNE(M)+1
    8      ZNVNI(M)=ZNVNI(M)+1
    9      TEMP1=ZNVNI(M)
   10      TEMP2=ZNVMNI(M)
   11      ZNVMNI(M)=MAXO(TEMP1,TEMP2)
   12      ZNVTIM(M)=ZNVTIM(M)-CLOCK
   13      CASEENTRY(M)
   14      CASE(2,3,4)
   15        VFELT(J)=CLOCK
   16      ENDCASE

   17    TRIP
   18    CASE(2)
   19      ZNTNE(M)=ZNTNE(M)+1
   20      ZNTNI(M)=ZNTNI(M)+1
   21      TEMP1=ZNTNI(M)
   22      TEMP2=ZNTMNI(M)
   23      ZNTMNI(M)=MAXO(TEMP1,TEMP2)
   24      ZNTTIM(M)=ZNTTIM(M)-CLOCK

   25      ZNPNE(M)=ZNPNE(M)+TPASS(J)
   26      ZNPNI(M)=ZNPNI(M)+TPASS(J)
   27      TEMP1=ZNPNI(M)
   28      TEMP2=ZNPMNI(M)
   29      ZNPMNI(M)=MAXO(TEMP1,TEMP2)
   30      ZNPTIM(M)=ZNPTIM(M)-CLOCK*TPASS(J)

   31      CASEENTRY(M)
   32      CASE(2,3,4)
   33        TFELT(J)=CLOCK
   34      ENDCASE
   35    ENDCASE
   36  ENDPROC
```

```
      PROJECT: ACT                    MEMBER: SZSTATES                    DATE:
      LIBRARY: DSM                    LEVEL:  01.05                       TIME:
      TYPE:    PCL                    USERID: C120496                     PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SZSTATES
    2     SZSTATES - ENTERING STATION LINK STATE
    3    VEHICLE/TRIP
    4    CASEENTRY(I)

    3    VEHICLE
    3    CASE(1)
    4      ZSVNE(M,N)=ZSVNE(M,N)+1
    5      ZSVNI(M,N)=ZSVNI(M,N)+1
    6      TEMP1=ZSVNI(M,N)
    6      TEMP2=ZSVMNI(M,N)
    5      ZSVMNI(M,N)=MAXU(TEMP1,TEMP2)
    5      ZSVTIN(M,N)=ZSVTIN(M,N)-CLOCK
    5      CASEENTRY(M)
    6      CASE(1)
    7        VARTSL(J)=CLOCK
    6      CASE(2)
    7        VEOLT(J)=CLOCK
    5      ENDCASE

    3    TRIP
    3    CASE(2)
    5      CALL ERROR(210,'SZSTAT WAS CALLED WITH SL & TRIP----PGM ERRO
    2   XR;',3)
    3    ENDCASE
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SZSTATET                    DATE
      LIBRARY: PSM                    LEVEL:  01.04                       TIME
      TYPE:    PCL                    USERID: C120496                     PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+---7-

    1  PROC:SZSTATET
    4     SZSTATET - ENTERING TRIP LINK STATE
    2    VEHICLE/TRIP
    3    CASEENTRY(I)

    3    VEHICLE
    3    CASE(1)
    5      CALL ERROR(211,'SZSTAT WAS CALLED WITH TL & VEH-----PGM ERRO
    2   R;',3)

    3    TRIP
    3    CASE(2)
    5      ZTTNE(M,N)=ZTTNE(M,N)+1
    5      ZTTNI(M,N)=ZTTNI(M,N)+1
    5      TEMP1=ZTTNI(M,N)
    5      TEMP2=ZTTMNI(M,N)
    3      ZTTMNI(M,N)=MAXC(TEMP1,TEMP2)
    3      ZTTTIN(M,N)=ZTTTIN(M,N)-CLOCK

    3      ZTPNE(M,N)=ZTPNE(M,N)+TPASS(J)
    5      ZTPNI(M,N)=ZTPNI(M,N)+TPASS(J)
    3      TEMP1=ZTPNI(M,N)
    5      TEMP2=ZTPMNI(M,N)
    3      ZTPMNI(M,N)=MAXC(TEMP1,TEMP2)
    3      ZTPTIN(M,N)=ZTPTIN(M,N)-CLOCK+TPASS(J)

    3      CASEENTRY(M)
    3      CASE(1)
    7        TARTTL(J)=CLOCK
    3      CASE(2)
    7        TFELT(J)=CLOCK
    3      ENDCASE
    3    ENDCASE
    1  ENDPROC
```

```
      PROJECT: AUT                          MEMBER: SZSTATL                          DATE
      LIBRARY: USM                          LEVEL:  01.04                            TIME
      TYPE:    PSL                          USERID: C120496                          PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7

    1  PROC: SZSTATL
    3    SZTATL - LEAVING STATE
    3      STATION/STATION LINK/TRIP LINK
    5      @CASEENTRY(K)

    5      STATION STATE
    5      @CASE(1)
    7        RUN SZSTATLN

    5      STATION LINK STATE
    5      @CASE(2)
    7        RUN SZSTATLS

    5      TRIP LINK STATE
    5      @CASE(3)
    7        RUN SZSTATLT

    5      @ENDCASE
    1  ENDPROC
```

```
     PROJECT: AGT                    MEMBER: SZSTATLN                    DATE
     LIBRARY: DSM                    LEVEL:  01.08                       TIME
     TYPE:    PLL                    USERID: CI20496                     PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC:SZSTATLN
   3    SZSTATLN - LEAVING STATION STATE
   3    VEHICLE/TRIP
   3    CASEENTRY(I)

   3    VEHICLE
   3    CASE(1)
   5      ZNVNL(M)=ZNVNL(M)+1
   5      ZNVNI(M)=ZNVNI(M)-1
   5      ZNVTIN(M)=ZNVTIN(M)+CLOCK
   5      CASEENTRY(M)
   5      CASE(1)
   7        DELTA=CLOCK-VARRT(J)
   5      CASE(2,3,4)
   7        DELTA=CLOCK-VFELT(J)
   5      ENDCASE
   5      ZNVSTL(M)=ZNVSTL(M)+DELTA
   5      TEMP1=ZNVMTL(M)
   5      ZNVMTL(M)=MAXO(TEMP1,DELTA)

   3    TRIP
   3    CASE(2)
   5      ZNTNL(M)=ZNTNL(M)+1
   5      ZNTNI(M)=ZNTNI(M)-1
   5      ZNTTIN(M)=ZNTTIN(M)+CLOCK

   5      CASEENTRY(M)
   5      CASE(1)
   7        DELTA=CLOCK-TARRT(J)
   5      CASE(2,3,4)
   7        DELTA=CLOCK-TFELT(J)
   5      ENDCASE

   5      ZNTSTL(M)=ZNTSTL(M)+DELTA
   5      TEMP1=ZNTMTL(M)
   5      ZNTMTL(M)=MAXO(TEMP1,DELTA)

   5      ZNPNL(M)=ZNPNL(M)+TPASS(J)
   5      ZNPNI(M)=ZNPNI(M)-TPASS(J)
   5      ZNPTIN(M)=ZNPTIN(M)+CLOCK*TPASS(J)

   5      ZNPSTL(M)=ZNPSTL(M)+DELTA
   5      TEMP1=ZNPMTL(M)
   5      ZNPMTL(M)=MAXO(TEMP1,DELTA)

   3    ENDCASE
   1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SZSTATLS                    DATE:
      LIBRARY: SSM                    LEVEL:  01.10                       TIME:
      TYPE:    PLL                    USERID: C120496                     PAGE:
START
 CCL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SZSTATLS
    2    SZSTATLS - LEAVING STATION LINK STATE
    2    VEHICLE/TRIP
    3    CASEENTRY(1)

    3    VEHICLE
    3    CASE(1)
    5      ZSVNL(M,N)=ZSVNL(M,N)+1
    5      ZSVNI(M,N)=ZSVNI(M,N)-1
    5      ZSVTIN(M,N)=ZSVTIN(M,N)+CLOCK
    5      CASEENTRY(M)
    5      CASE(1)
    7        DELTA=CLOCK-VARTSL(J)
    5      CASE(2)
    7        DELTA=CLOCK-VDELT(J)
    5      CASE(3)
    7        DELTA=CLOCK-VTIME(J)/10
    5      ENDCASE
    5      ZSVSTL(M,N)=ZSVSTL(M,N)+DELTA
    5      TEMP1=ZSVMTL(M,N)
    5      ZSVMTL(M,N)=MAX0(TEMP1,DELTA)

    3    TRIP
    3    CASE(2)
    5      CALL ERROR(210,'SZSTAT WAS CALLED WITH SL & TRIP----PGM          ERR
   4  OR;',0)
    3    ENDCASE
    1  ENDPROC
```

```
      PROJECT: AGT                          MEMBER: SZSTATLT                        DATE
      LIBRARY: LSM                          LEVEL:  01.09                           TIME
      TYPE:    PLL                          USERID: CI20496                         PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SZSTATLT
    3    SZSTATLT - LEAVING TRIP LINK STATE
    3    VEHICLE/TRIP
    3    CASEENTRY(I)

    3    VEHICLE
    3    CASE(1)
    3      CALL ERROR(211,'SZSTAT WAS CALLED WITH TL & VEH-----PGM          ERR
    2   OR;',3)

    3    TRIP
    3    CASE(2)
    5      ZTTNL(M,N)=ZTTNL(M,N)+1
    5      ZTTNI(M,N)=ZTTNI(M,N)-1
    5      ZTTTIN(M,N)=ZTTTIN(M,N)+CLOCK
    5      CASEENTRY(M)
    5      CASE(1)
    7        DELTA=CLOCK-TARTTL(J)
    5      CASE(2)
    7        DELTA=CLOCK-TFELT(J)
    5      CASE(3)
    7        DELTA=CLOCK-TTIME(J)/10
    5      ENDCASE
    5      ZTTSTL(M,N)=ZTTSTL(M,N)+DELTA
    5      TEMP1=ZTTMTL(M,N)
    5      ZTTMTL(M,N)=MAXO(TEMP1,DELTA)

    5      ZTPNL(M,N)=ZTPNL(M,N)+TPASS(J)
    5      ZTPNI(M,N)=ZTPNI(M,N)-TPASS(J)
    5      ZTPTIN(M,N)=ZTPTIN(M,N)+CLOCK*TPASS(J)

    5      ZTPSTL(M,N)=ZTPSTL(M,N)+DELTA
    5      TEMP1=ZTPMTL(M,N)
    5      ZTPMTL(M,N)=MAXO(TEMP1,DELTA)
    3    ENDCASE
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: SZZERO                    DATE:
      LIBRARY: SRM                    LEVEL:  01.00                     TIME:
      TYPE:    PEL                    USERID: CIL0496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:SZZERO
    2    ZERO COUNTS
    3    RESET MAXIMA TO CURRENT VALUES OF COUNTS
    4    RESET INTEGRAL BY SUBTRACTING END POINT VALUE
    1  ENDPROC
```

```
     PROJECT: AGT                    MEMBER: ZABIN                    DATE:
     LIBRARY: DSM                    LEVEL:  01.00                    TIME:
     TYPE:    PUL                    USERID: H639547                  PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC ABIN <CHECK IF SUFFICIENT SPACE HAS BEEN ALLOTTED TO A BIN.
  12            IF NOT, IT PROVIDES THE CHANGES NECESSARY TO PROVIDE THE
  12            REQUIRED BIN SPACE. EITHER LEAVE THE BIN UNCHANGED
  12            OR INCREASE THE SIZE OF IT, LEAVING PREVIOUS CONTENTS OF BIN
  12            UNCHANGED; IT WILL NEVER DECREASE THE SIZE OF THE BIN.
   8        PART GIVES THE NUMBER OF ADDITIONAL SPACES REQUIRED.>
   3    IF
   5      AT REMAINING SPACE >0
   3    THEN
   5      BIN IS LARGE ENOUGH
   3    ELSE < PROVIDE THE REQUIRED BIN SPACE>
   5      IF
   6         BLANK AREA ADJACENT TO BIN AND BLANK+OLD LARGE ENOUGH
   3      THEN
   7        EXPAND BIN TO INCLUDE BLANK AREA
   7        IF
   1  C       MORE SPACE THAN REQUIRED
   7        THEN
   9          SET UP REMAINING BLANK SPACE AS A SEPARATE BIN
   7        ENDIF
   5      ELSE
   7        IF
  10           BLANK AREA WAS END-OF-BIN-AREA
   7        THEN
   9          POINT INDEX IN BIN TO AREA
   7        ELSE <BIN EXPANSION NOT POSSIBLE WITHOUT SHIFTING DATA AROUND>
   9          IF
  11            ENOUGH SPACE AT END OF ALLOTTED BIN AREA.
   9          THEN
  11            GETUP BIN MARKER & ID FOR NEW AREA
  11            MOVE CONTENTS OF THE BIN FROM OLD AREA TO NEW
  11            ZERO OUT ALL OF OLD AREA EXCEPT FOR BIN SIZE INDICATOR
   3         ELSE <NOT ENOUGH SPACE AT END OF ALLOTED AREA>
  11            RUN-SHIFT <MOVE UP BINS TO COLLECT SPACE INTO ONE LARGE BIN
  13              AT THE END OF THE ENTIRE BIN AREA
  11            IF
  13              BLANK AREA TOO SMALL
  11            THEN
  13              ERROR <NO SPACE>
  11            ELSE
  13              MOVE ALL BINS BELOW NB DOWN ENOUGH FOR NEW NB SIZE
  13              IF
  15                CURRENT BIN AND BLANK AREA ADJACENT
  13              THEN
  15                CHANGE ORIGIN OF BLANK AREA
  13              ELSE
  15                REDEFINE BLANK AREA
  14                MOVE ALL BINS BELOW NB DOWNWARD ACCORDING TO REQD SPACE
  14                CHANGE BIN LOC POINTERS FOR ALL MOVED BINS
  12              ENDIF
```

```
      PROJECT: ACF                    MEMBER: ZAGIN                    DATE
      LIBRARY: DSN                    LEVEL:  01.00                    TIME
      TYPE:    PSL                    USERID: H639547                  PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7.

  10            END IF
   8          END IF
   6        END IF
   4      END IF
   2    END-DO
```

```
      PROJECT: AGT                    MEMBER: ZBINL                    DATE
      LIBRARY: USM                    LEVEL:  01.00                    TIME
      TYPE:    PDL                    USERID: CI20496                  PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:ZBINL  <GET LENGTH OF DATA IN BIN>
    3    COMPUTE LENGTH FROM BIN HEADER INFO
    1  ENDPROC
```

```
    PROJECT: AGI                    MEMBER: ZBNCHK                    DATE:
    LIBRARY: USR                    LEVEL:  00.01                     TIME:
    TYPE:    PCL                    USERID: NASDTC                    PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC BNCHK <BIN ALLOCATION PROCESSING>
   6       <BNCHK CHECKS IF SUFFICIENT SPACE HAS BEEN ALLOCATED TO A NEW BIN.
   7        IF ADDITIONAL SPACE IS REQUIRED FOR THE BIN, THE CURRENT BIN
   7        ASSIGNMENT IS CANCELLED & THE BIN IS REDEFINED IN AN AREA WITH
   7        SUFFICIENT SPACE. BNCHK ALSO PROVIDES FOR THE EXPANSION OF ACTIVE
   7        BINS IF REQUIRED TO SUPPORT ADDITIONAL DATA STORAGE.>

   3    IF <NEW BIN IS REQUIRED>
   3      REQTBL(REQNO,1)=0
   3    THEN <PERFORM BIN ALLOCATION>
   5      IF
   7        THERE IS SUFFICIENT SPACE AT END OF ALLOTTED BIN AREA
   5      THEN
   7        SET BIN SIZE AND BIN ID
   7        SET ID IF BIN AREA PREVIOUSLY ALLOCATED TO THIS BIN TO ZERO
   7        SET BIN LOCATION MARKER
   7        SET START MARKER
   7        SET ID AND REMAINING ENTRIES OF OLD BIN AREA TO ZERO
   7        SET ALL UNUSED SPACES OF NEW AREA TO ZERO
   5      ELSE       <THERE IS NOT ENOUGH SPACE>
   7        SET ID:=0  <THROW AWAY BIN THAT IS TOO SMALL>
   7        PUSH UP AREAS IN USE TO COLLECT UNUSED BIN SPACES INTO ONE
   7        LARGE AREA WITH ID:=0 AT THE END OF THE ENTIRE BIN AREA
   7        IF
   9          THERE IN NOT ENOUGH NEW SPACE
   7        THEN
   9          WRITE ERROR---NO SPACE
   9          RETURN
   7        ELSE
   9          SET BIN SIZE AND BIN ID
   9          SET BIN LOCATION MARKER
   9          SET START MARKER
   7        ENDIF
   5      ENDIF
   3    ELSE <PROCESS EXPANSION OF A CURRENT BIN IN USE>
   3    <CHECK IF SUFFICIENT SPACE HAS BEEN ALLOTTED TO BIN. IF NOT SUFFIC-
   4     IENT, PROVIDE THE CHANGES NECESSARY TO PROVIDE THE REQUIRED SPACE.
   6       EITHER LEAVE THE BIN UNCHANGED OR INCREASE ITS SIZE, LEAVING THE
   6       CONTENTS OF THE BIN UNCHANGED>
   5      PART GIVES THE NUMBER OF ADDITIONAL SPACES REQUIRED FOR THE BIN>
   5      IF
   8        BIN AS CURRENTLY DEFINED IS =LARGE ENOUGH
   5      THEN <PROVIDE THE REQUIRED BIN SPACE>
   7        IF
   9          BLANK AREA ADJACENT TO BIN AND BLANK+OLD LARGE ENOUGH?
   7        THEN <GRAB SPACE FROM THAT BLANK AREA>
   9          DIF:=END OF ADJACENT AREA-END OF CURRENT BIN
   9          IF <ENOUGH BLANK AREA LEFT OVER FOR AN EMPTY BIN>
  11            DIF>=4
   9          THEN
```

```
      PROJECT: ACT                    MEMBER: ZBNCHK                    DATE:
      LIBRARY: DSM                    LEVEL:  00.01                     TIME:
      TYPE:    PDL                    USERID: NA3DTC                    PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  11              USE EXCESS FOR ANOTHER BIN
   9            ENDIF
   7          ELSE
   5            FIND AREA IN BIN STORAGE WITH ENOUGH SPACE
   9            IF
  11              BIN EXPANSION NOT POSSIBLE WITHOUT SHIFTING DATA AROUND
   9            THEN
  11              CHECK IF THERE IS ENOUGH SPACE AT END OF ALLOTTED BIN AREA
  11              IF
  13                SUFFICIENT SPACE AT END OF ALLOTTED BIN AREA
  11              THEN
  13                SET BIN SIZE & ID FOR NEW AREA
  13                MOVE CONTENTS OF BIN TO NEW AREA
  13                RESET BIN LOCATION POINTER
  11              ELSE
  13                PUSH UP AREAS IN USE TO COLLECT ALL UNUSED SPACE IN ONE
  13                AREA AT END OF BIN STORAGE
  15                IF
  13                  SUFFICIENT SPACE IN FREE AREA
  15                THEN
  13                  EXPAND BIN
  15                ELSE
  13                  ERROR - NO SPACE IN BIN STORAGE
  15                ENDIF
  11              ENDIF
   9            ENDIF
   7          ENDIF
   5        ENDIF
   3      ENDIF
   1    ENDPROC
```

```
      PROJECT: AGT                    MEMBER: JDSIN                    DATE:
      LIBRARY: SRM                    LEVEL:  01.00                    TIME:
      TYPE:    PSL                    USERID: H639547                  PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC DSIN <PERFORM INITIAL BIN AREA ALLOCATION & SETUP>
    3    DO <PERFORM INITIAL BIN ALLOCATION & SETUP>
    5      LOC:=0 <POINTER TO FIRST BIN IN STORAGE AREA>
    5      FOR <MAXIMUM ALLOWABLE BINS>
    7        BINR=1 TO KYBINS
    5      DO <INITALIZE BIN LOCATION POINTERS & ALLOCATE STORAGE>
    7        BPTR(BINR):=LOC
    7        BAREA(LOC-2):=5  <INITIAL BIN LENGTH>
    7        BAREA(LOC-1):=BINR <BIN ID>
    7        BAREA(LOC):=LOC
    7        BAREA(LOC+1):=LOC+1 <INITIAL DATA INDEX>
    7        BAREA(LOC+2):=LOC+1 <INITAL END OF BIN DATA>
    7        LOC:=LOC+5
    5      ENDDO
    5      DO <DEFINE REST OF STORAGE AREA AS PSEUDO BIN>
    7        BPTR(BINR):=LOC
    7        BAREA(LOC-2):=5  <INITIAL BIN LENGTH>
    7        BAREA(LOC-1):=BINR <BIN ID>
    7        BAREA(LOC):=LOC
    7        BAREA(LOC+1):=LOC+1 <INITIAL DATA INDEX>
    7        BAREA(LOC+2):=KMSIZE <INITAL END OF BIN DATA>
    5      ENDDO
    3    ENDDO
    1  ENDPROC
```

```
      PROJECT: AGI                    MEMBER: ZDUMBIN                 DATE:
      LIBRARY: JSM                    LEVEL:  01.00                   TIME:
      TYPE:    PLL                    USERID: C120496                 PAGE:
START
 COL  ----+----.----+----2----+----3----+----4----+----5----+----6----+----7-

   1  PROC:ZDUMBIN <DUMP CONTENTS OF BIN STORAGE AREA>
   3    PRINT DATA FROM BIN HEADERS
   1  ENDPROC
```

```
      PROJECT: ACT                    MEMBER: ZERROR                    DATE:
      LIBRARY: DSM                    LEVEL:  01.00                     TIME:
      TYPE:    PLC                    USERID: C120496                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC ZERROR <WRITE ERROR MESSAGE & CONTINUE/TERMINATE>
    2    DETERMINE LENGTH OF MESSAGE(LOOK FOR SEMICOLIN)
    3    PRINT MESSAGE TEXT
    3    PRINT STANDARD LINE ASSOCIATED WITH SEVERITY
    3    PRINT VALUE OF CLOCK
    3    INCREMENT NUMBER OF MESSAGES COUNTERS
    3    IF
    5      MESSAGE TYPE IS SEVERE                                      OR
    5      NUMBER OF INFORMATIVE MESSAGES > KMMSGI                     OR
    5      NUMBER OF WARNING MESSAGES > KMMSGW                         OR
    5      NUMBER OF INFORMATIVE + WARNING MESSAGES > KMMSGS           OR
    5      NUMBER OF MESSAGES OF ANY GIVEN ID > KMMTYP
    3    THEN
    5      TERMINATE SIMULATION
    3    ENDIF
    1  ENDPROC
```

```
      PROJECT: AQI                    MEMBER: ZFLAG                    DATE:
      LIBRARY: USM                    LEVEL:  01.00                    TIME:
      TYPE:    PLL                    USERID: C120496                  PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC:/FLAG <RESET ALL DEBUG FLAGS>
    2    SET ALL FLAG VALUES FALSE
    3    UNTIL
    5      A ZERO FLAG FIELD FOUND ON A CARD
    6    DO
    8      SET STRINGS OF CONSECUTIVE FLAGS TRUE
    9    ENDDO
    1  ENDPROC
```

```
      PROJECT: AOI                    MEMBER: ZGRAPH                    DATE:
      LIBRARY: DSM                    LEVEL:  00.01                     TIME:
      TYPE:    PLI                    USERID: H639547                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--

   1  PROC GRAPH <PRODUCE A SERIES PLOT OF SAMPLED DATA VALUES>
  12             <FOR X VALUES--X0,X0+DELTAX,...,X0+NDELTA*DELTAX
  13              A PLOT OF NY DEPENDENT VARIABLES--Y1,Y2,...--IN
  13              THE LINEAR RANGE (BOTTOM,TOP) IS CREATED & DISPLAYED>
   3    FOR
   5      I=1,101             <SETUP THE GRID>
   3    DO
   5      GRID(I):=DASH
   5      PLOT(I):=BLANK
   3    ENDDO
   3    FOR
   5      I=1,101,BY 10
   3    DO
   5      GRID(I):=CROSS
   5      PLOT(I):=CROSS
   3    ENDDO
   3    IF
   5      SCALE NOT ENTERED
   3    THEN                    <COMPUTE SCALE>
   5      FIND MINIMUM AND MAXIMUM VALUES OF Y
   5      BOTTOM:=YMIN
   5      TOP:=YMAX
   5      SCALE:=(TOP-BOTTOM)/100
   3    ENDIF
   3    MOVE VALUES IN BIN TO SLOT IN GRID
   3    WRITE TITLE
   3    WRITE LABELS
   3    FOR
   5      # OF LINES IN GRID
   3    DO                      <WRITE OUT THE GRID>
   5      WRITE YVALUE,XVALUE,GRIDLINE
   3    ENDDO
   1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: ZHEADER                    DATE:
      LIBRARY: CSM                    LEVEL:  00.01                      TIME:
      TYPE:    PLL                    USERID: H639547                    PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--

    1  PROC HEADER <READ A HEADER RECORD FROM THE RAW STATISTICS FILE>
    2     READ NEXT RECORD
    3     IF
    4       NAME = HEADER
    5     THEN <AN ERROR EXISTS>
    6       ERROR <EXPECTED HEADER RECORD NOT FOUND>
    7       DOUNTIL <LOCATE NEXT HEADER IN FILE>
    8         NAME = HEADER
    9         READ NEXT RECORD
   10       ENDDO
   11  ENDPROC
```

```
      PROJECT: AOI                    MEMBER: ZHIST                    DATE:
      LIBRARY: USM                    LEVEL:  00.01                    TIME:
      TYPE:    PLL                    USERID: H639547                  PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--

    1  PROC ZHIST <PRODUCE A HISTOGRAM OF SAMPLED ITEMS>
    2    RUN MMMX <DETERMINE MINIMUM & MAXIMUM VALUES OF DATA ITEMS>
    4    RUN ZNCHK <ENSURE BIN CAN HOLD COMPUTED VALUES>
    5    RUN HIST <DISPLAY HISTOGRAM>
    1  ENDPROC
```

```
      PROJECT: AUT                    MEMBER: ZLIST                    DATE:
      LIBRARY: USR                    LEVEL:  00.02                    TIME:
      TYPE:    PLL                    USERID: MG3954?                  PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC ZLIST <LIST ALL SAMPLE VALUES OR DISPLAY SAMPLE SUMMARY>
   15                <ZLIST PRINTS OUT THE CONTENTS OF ANY
   16                 SPECIFIED BIN. LISTING EVERY KTH ELEMENT. THE BIN
   16                 TO BE PRINTED IS BIN(1). K IS GIVEN BY PAR(1).
   16                 IF EVERY ELEMENT IS TO BE LISTED, PAR(1) MAY BE
   16                 LEFT BLANK AND IT IS ASSUMED THAT PAR(1)=1.      >
    3    IF           <PAR(1) HAS BEEN LEFT BLANK>
    6      PAR(1)=0
    3    THEN         <SET PAR(1)=1>
    6      PAR(1)=1
    3    ENDIF
    3    IF
    5      BINS ARE EMPTY
    3    THEN
    5      ERROR <BINS ARE EMPTY>
    6      RETURN
    3    ENDIF
    3    RUN LIST
    1  ENDPROC
```

```
      PROJECT: AST                    MEMBER: ZMINMX                    DATE:
      LIBRARY: DSM                    LEVEL:  00.01                     TIME:
      TYPE:    -CL                    USERID: H639547                   PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--

    1  PROC MNMX <DETERMINE THE MINIMUM, MAXIMUM, AND RANGE OF THE VALUES IN A
   12              BIN (NBIN1) & STORE THEM IN THE BIN>
    3    FOR
    5      EACH ITEM IN THE BIN
    3    DO <FIND MINIMUM VALUE AND STORE>
    5      IF
    7        VALUE < OLD MIN
    5      THEN
    7        STORE CURRENT VALUE AS MIN
    5      ENDIF
    3    ENDDO
    3    FOR
    5      EACH ITEM IN THE BIN
    3    DO <FIND MAXIMUM VALUE AND STORE>
    5      IF
    7        VALUE < OLD MAX
    5      THEN
    7        STORE CURRENT VALUE AS MAX
    5      ENDIF
    3    ENDDO
    3    DO <DETERMINE RANGE OF VALUES AND STORE>
    5      RANGE=MAX-MIN
    5      STORE RANGE IN BIN
    3    ENDDO
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: ZRCLEAN                 DATE:
      LIBRARY: DSM                    LEVEL:  00.01                   TIME:
      TYPE:    PLL                    USERID: M6C9547                 PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--

    1  PROC ^CLEAN <RESET BIN AREA POINTERS>
    3    DO <FIXUP BIN STOP ITEM>
    5      LAST ITEM INDEX = FIRST ITEM INDEX
    4    UNTIL
    6      ALL BIN AREAS RESET
    4    ENDDO
    2  ENDPROC
```

```
PROJECT: ASI                    MEMBER: ZREQU                    DATE:
LIBRARY: USM                    LEVEL:  00.00                    TIME:
TYPE:    PLL                    USERID: NASMH                    PAGE:
START
COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

  1  PROC ZREQU <FILE A USER DATA REQUEST>
  3    SIZE:=100
  5    IF <MAX NUMBER OF REQUESTS FILED>
  5      REQNO=MXREQ
  3    THEN
  5      ERROR - TOO MANY REQUESTS
  3    ELSE <FILE NEW REQUEST>
  5      REQNO:=REQNO+1
  5      DO <PERFORM BIN ALLOCATION>
  7        IF <HISTIGRAM DISPLAY REQUESTED>
  9          FTYPE=1 <HISTOGRAM>
  7        THEN
  9          SIZE:=20
  7        ENDIF
  7        REQTBL(REQNO):=SIZE
  7        BIN:=BIN+1
  7        RUN ENCHK <ALLOCATE BIN SPACE>
  5      ENDDO
  5      DO <CREATE REQUEST ENTRY>
  7        REQTBL(1):=BIN    <BIN ASSIGNED TO REQUEST>
  7        REQTBL(2):=SIZE   <INITIAL BIN SIZE>
  7        REQTBL(3):=MCAT   <MAIN CATAGORY MNEMONIC>
  7        REQTBL(4):=SCAT   <SUBCATAGORY MNEMONIC>
  7        BINR:=BYTR(BIN)
  7        BINR:=BINR+1
  7        REQTBL(5):=BINR   <FIRST AVAILABLE POSITION IN BIN>
  7        REQTBL(6):=SIZE   <REMAINING BIN SPACE>
  7        REQTBL(7):=0      <CHAIN WORD>
  5      ENDDO
  3    ENDIF
  1  ENDPROC
```

```
      PROJECT: AGI                    MEMBER: ZSHIFT                  DATE:
      LIBRARY: USR                    LEVEL:  00.01                   TIME:
      TYPE:    PLI                    USERID: H639547                 PAGE:
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC SHIFT <MOVE AREAS WITH ID=0 UP,ONE AT A TIME UNTIL A BIN IN
   10            USE IS ENCOUNTERED.  ALL ENTRIES OF THE OLD AREA ARE SET TO 0>
    3    FOR
    5      EACH BIN IN ALLOCATED AREA
    3    DO
    5      MOVE BIN UP IN STORAGE AREA
    3    UNTIL
    5      BIN IN USE ENCOUNTERED
    3    ENDDO
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: ZSKIPFO                    DATE:
      LIBRARY: LSM                    LEVEL:  00.01                      TIME:
      TYPE:    PCL                    USERID: F639547                    PAGE:
START
 CDL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC ZSKIPFO <SKIP FOLLOWER RECORDS IN RAW STATISTICS FILE>
    3    IF
    5      NUMBER OF FOLLOWERS>0
    6      DO<SKIP RECORDS>
    7        READ NEXT RECORD
    7        IF
    9          NAME == FOLLOW
    7        THEN <AN ERROR EXISTS>
    9          ERROR <EXPECTED FOLLOWER DOES NOT EXIST>
    7        ENDIF
    5      UNTIL
    7        REQUIRED NUMBER OF FOLLOWERS SKIPPED
    5      ENDDO
    1  ENDPROC
```

```
      PROJECT: AGT                    MEMBER: ZSTORE                    DATE
      LIBRARY: USM                    LEVEL:  00.01                     TIME
      TYPE:    PLL                    USERID: mc39547                   PAGE
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-

    1  PROC STORE <STORE A NEW PIECE OF DATA INTO SIN GIVEN BY L>
    2    IF
    3      TIME SERIES OUTPUT REQUIRED
    4    THEN <TIME SERIES STORAGE - DECK SPACE LEFT, OK IF MORE NEEDED>
    5      ZREQUC(8,L)=ZREQUC(8,L)-1
    6      IF <GET MORE SPACE REQUIRED>
    7        ZREQUC(8,L)<=0
    8      THEN
    7        RUN ASIN
    7        RESET LOCATION POINTERS INTO SIN ARRAY
    7        BUMP LAST LOC STORED POINTER & STORE ITEM
    5      ENDIF
    3    ELSE <STATISTICAL SUMMARY STORED INTO SIN>
    5      FOR
    7        MIN & MAX VALUES STORED IN SIN
    8      DO
    7        IF
    4          NEW DATA < OLD MIN
    7        THEN
    9          STORE VALUE AS MIN
    9          RECORD TIME OF MIN IN SIN
    7        ENDIF
    7        IF
    9          NEW DATA > OLD MAX
    7        THEN
    9          STORE VALUE AS NEW MAX
    4          RECORD TIME OF MAX IN SIN
    7        ENDIF
    8      ENDDO
    3    ENDIF
    1  END.PROC
```

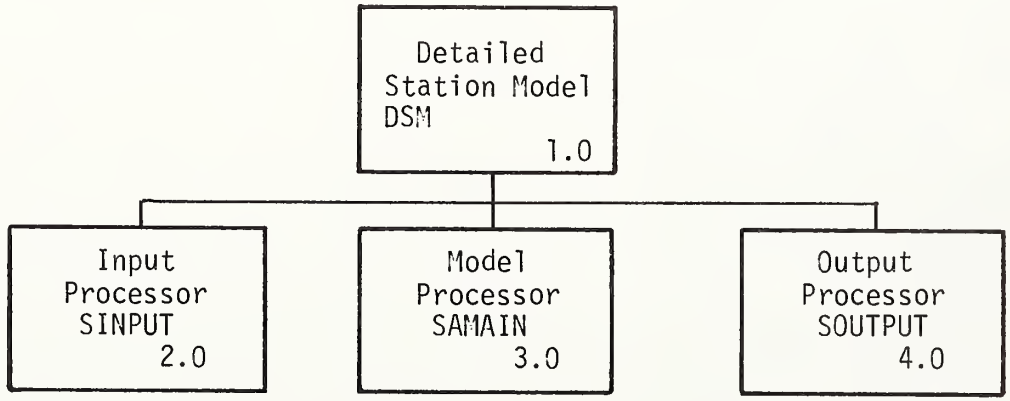
APPENDIX B

HIPO DIAGRAMS

The diagrams in this appendix illustrate the DSM high-level design through the use of Hierarchy plus Input-Process-Output (HIPO) diagrams. The Visual Table of Contents illustrates program organization and contains the names and identification numbers of the detail Input-Process-Output diagrams that define the processing to be performed. These diagrams should be used in conjunction with the Process Design Language (PDL) descriptions contained in Appendix A, which provide extended descriptions of the program design in greater detail. Where the Visual Table of Contents and Input-Process-Output diagrams reference a segment name and indentification number, that segment is further expanded in both an Input-Process-Output diagram (having that identification number) and a PDL segment (having that segment name). If an Input-Process-Output diagram references a function by segment name only, then the design of that segment will be found in the PDL segment having that segment name. These HIPO diagrams are intended to supply a high-level introductory description of the processing; PDL and component descriptions provide the detail.

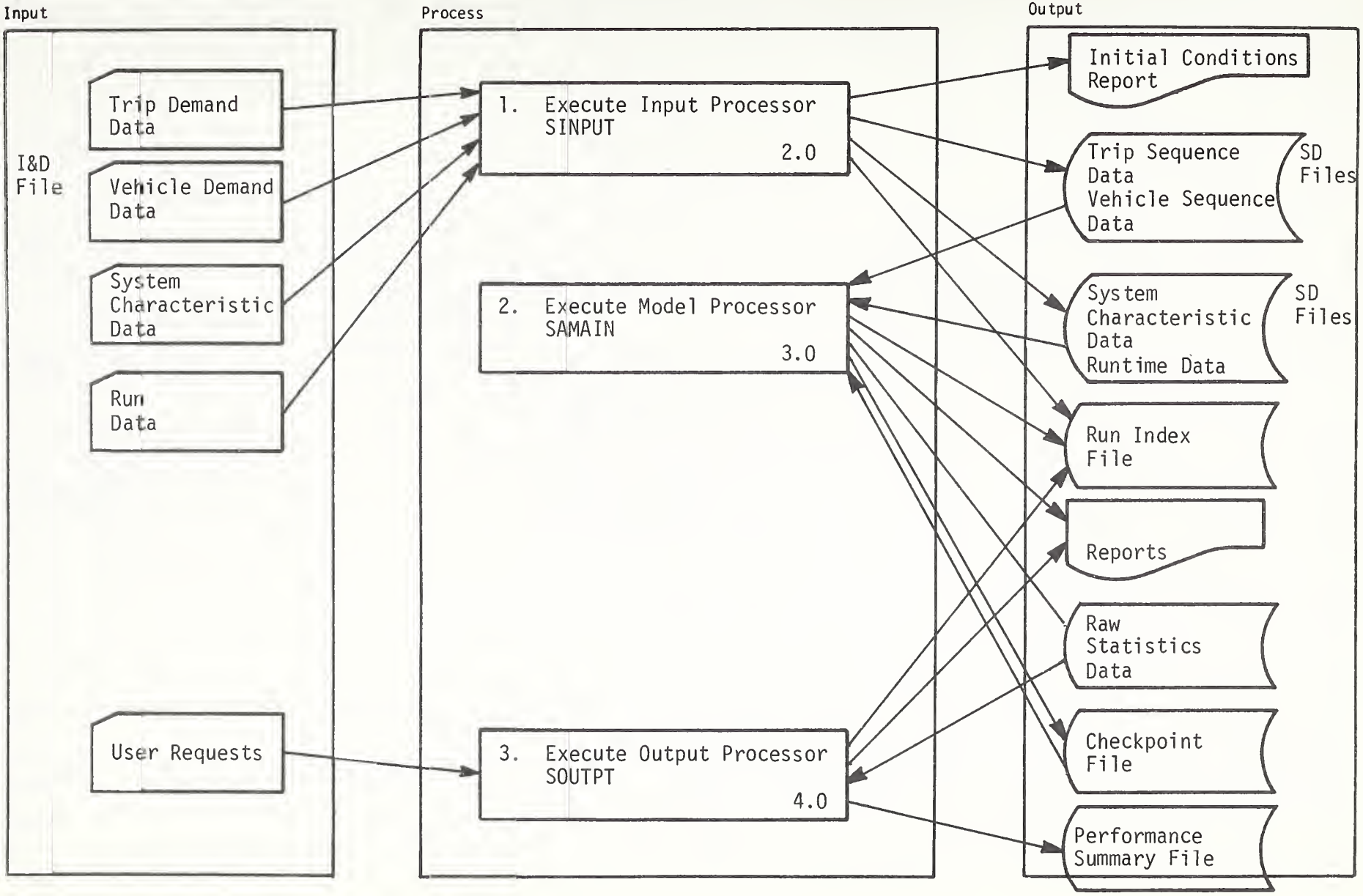
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: VTOC Name: Visual Table of Contents Description: Detailed Station Model



Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

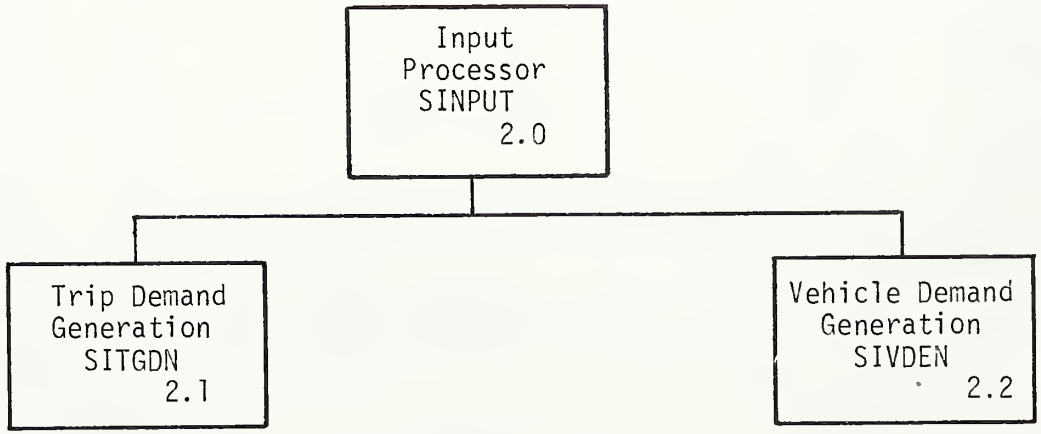
Diagram ID: 1.0 Name: DSM Description: Detailed Station Model



Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

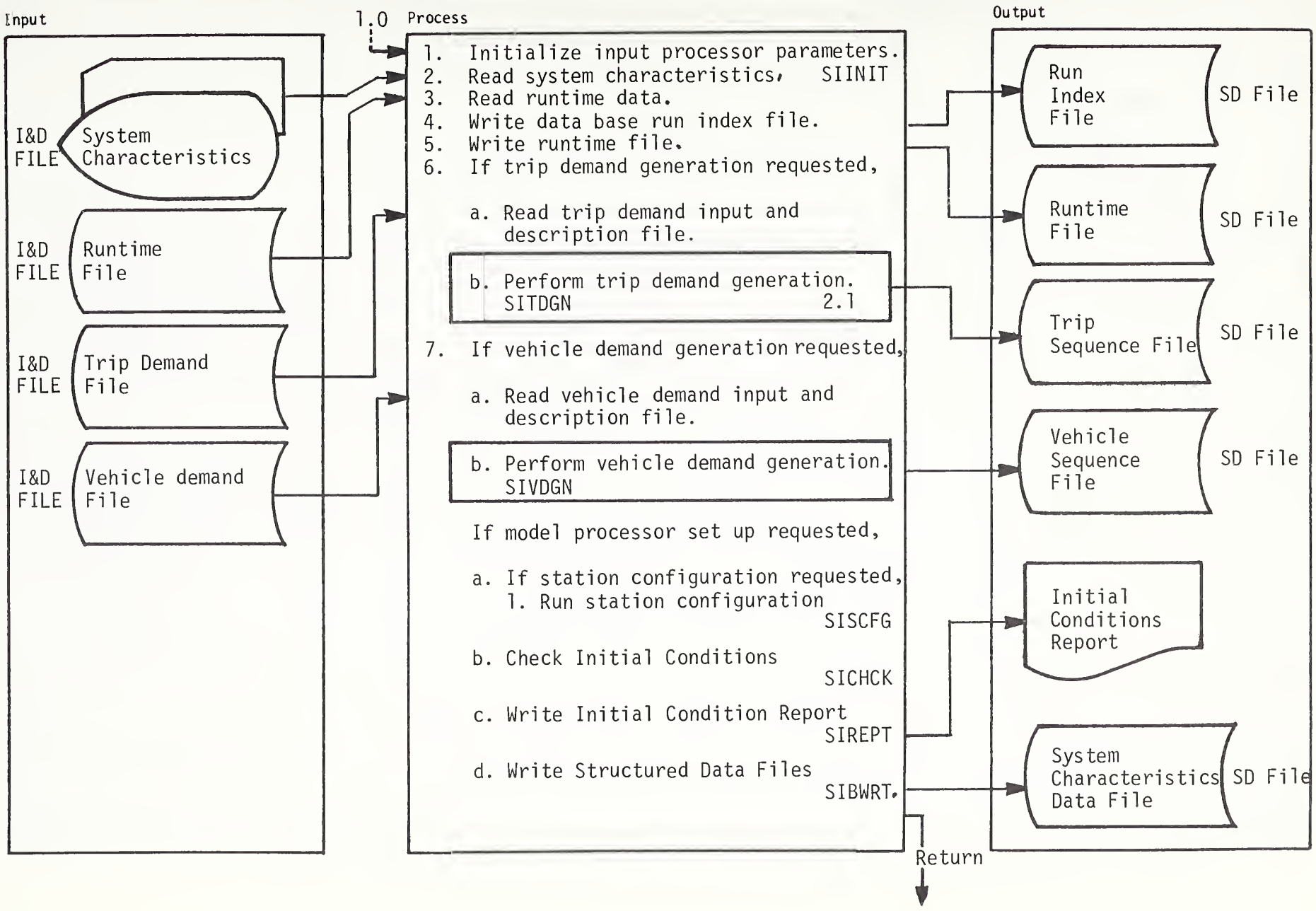
Diagram ID: VTOC Name: Visual Table of Contents Description: Input Processor

Input | Process | Output



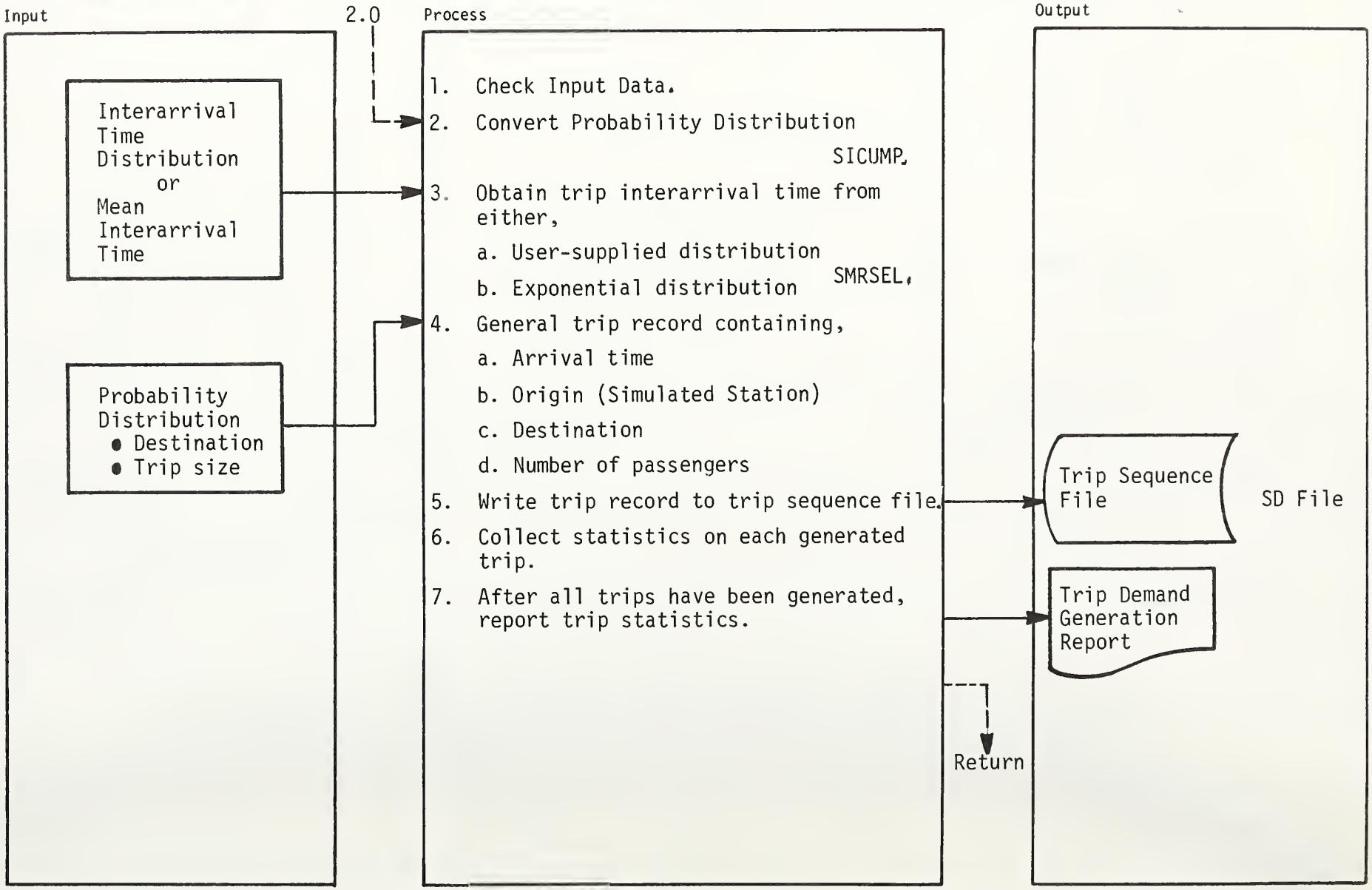
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 2.0 Name: SINPUT (Section 2.1.2.1) Description: Input Processor



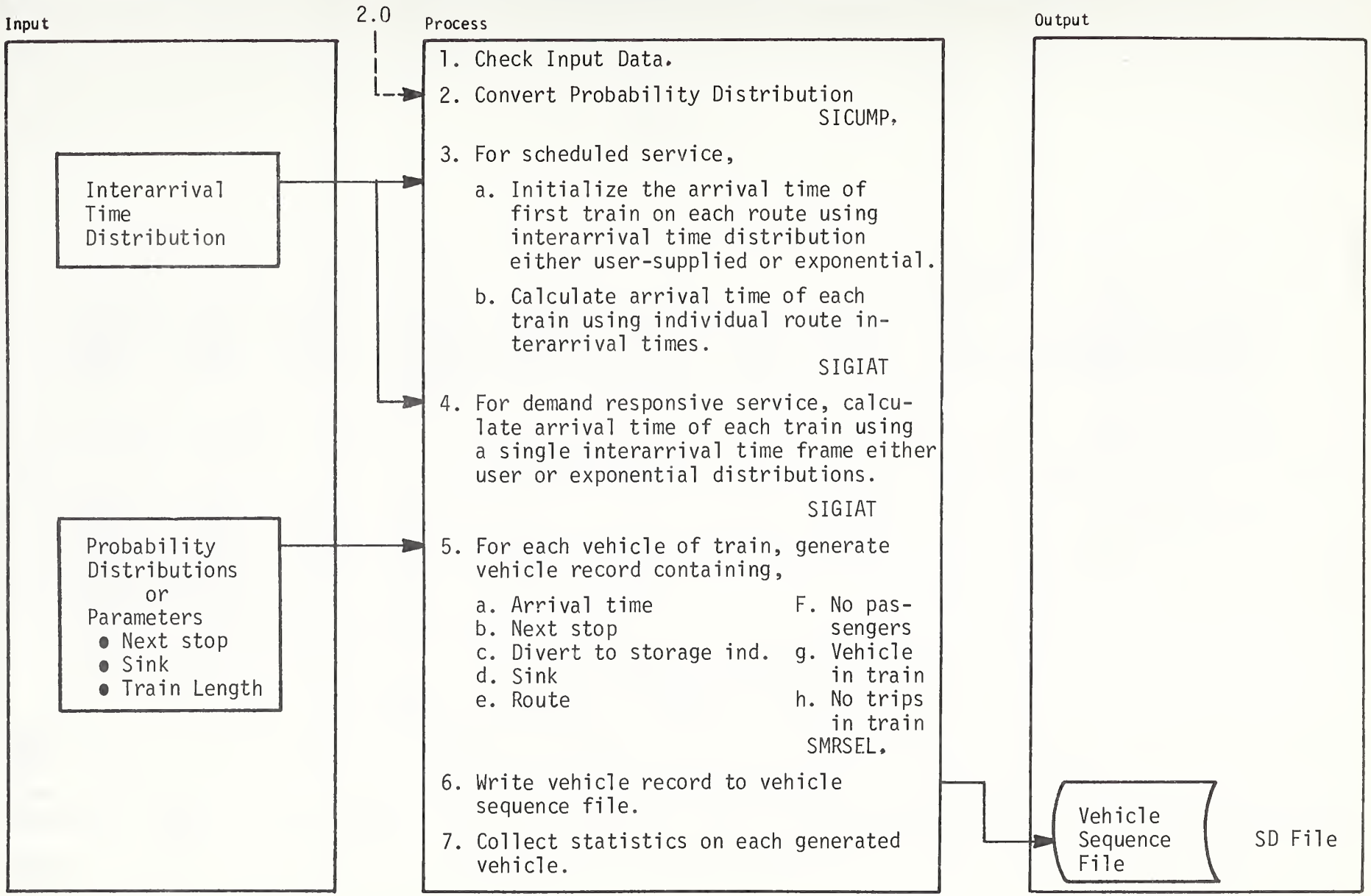
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 2.1 Name: SITDGN (Section 2.1.2.9) Description: Trip Demand Generation



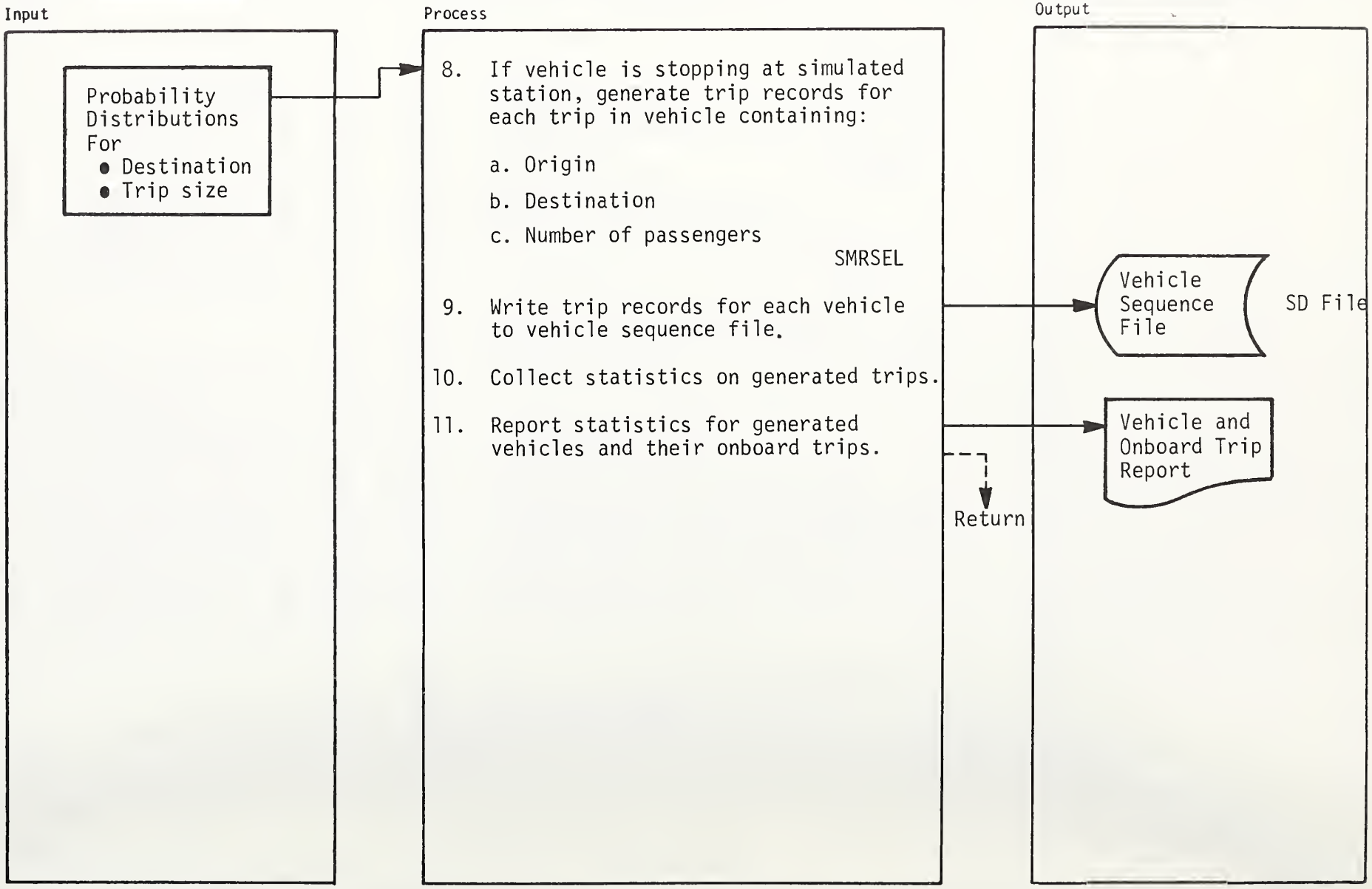
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 2

Diagram ID: 2.2 Name: SIVDGN (Section 2.1.2.10) Description: Vehicle Demand Generation



Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 2 of 2

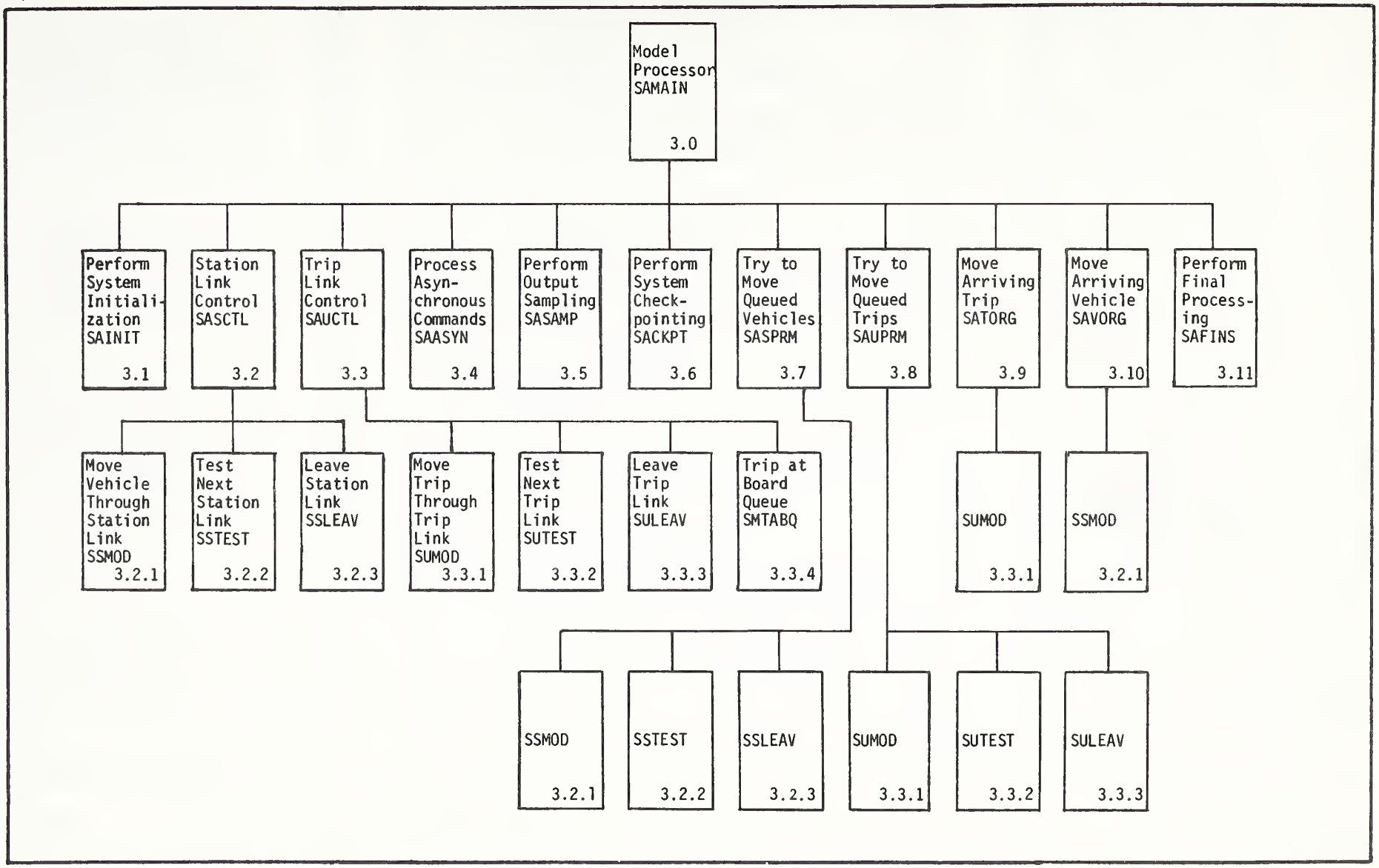
Diagram ID: 2.2 Name: VDG (Section 2.1.2.3) Description: Vehicle Demand Generation



Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

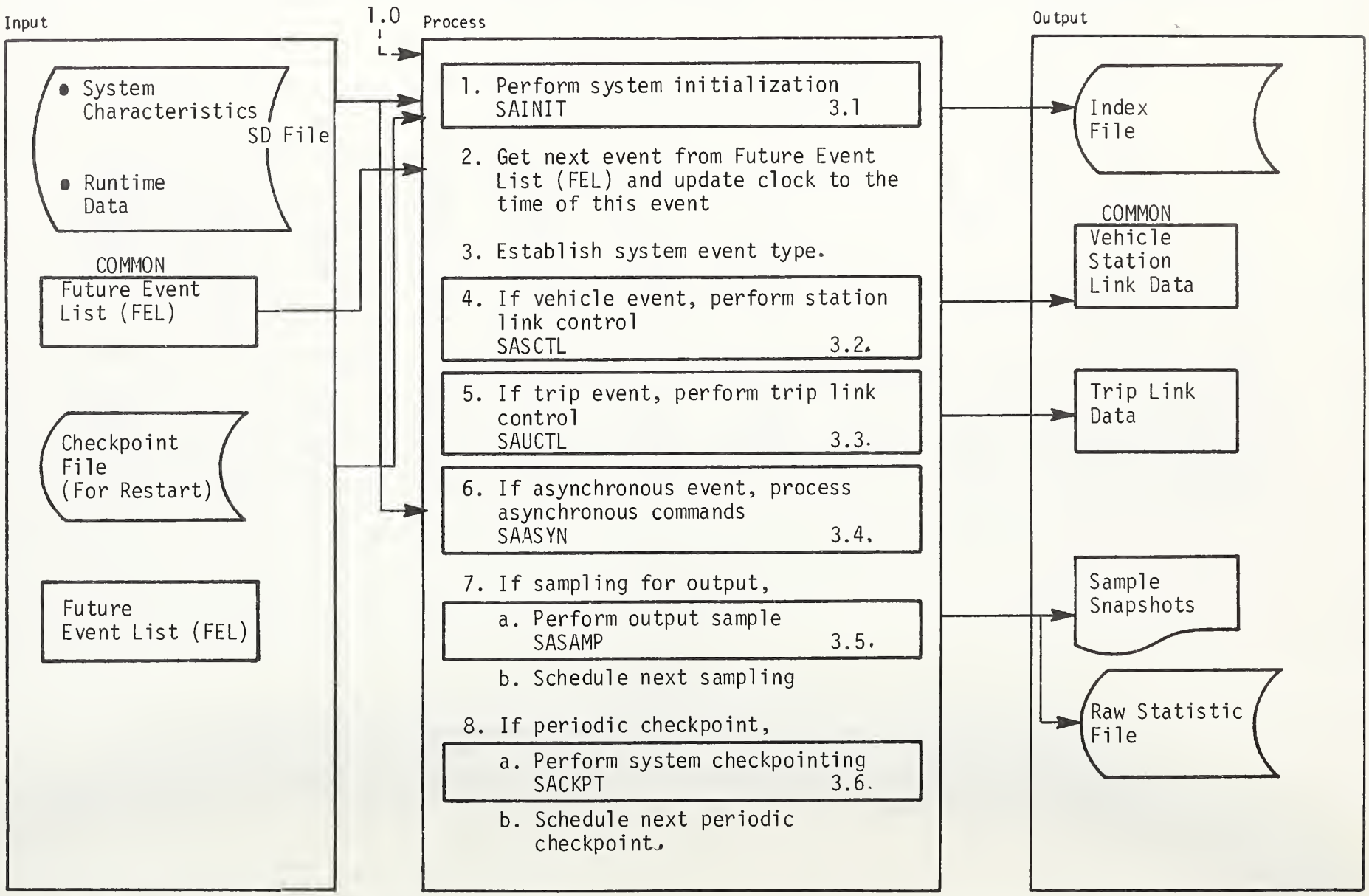
Diagram ID: VTOC Name: Visual Table of Contents Description: Model Processor

Input Process Output



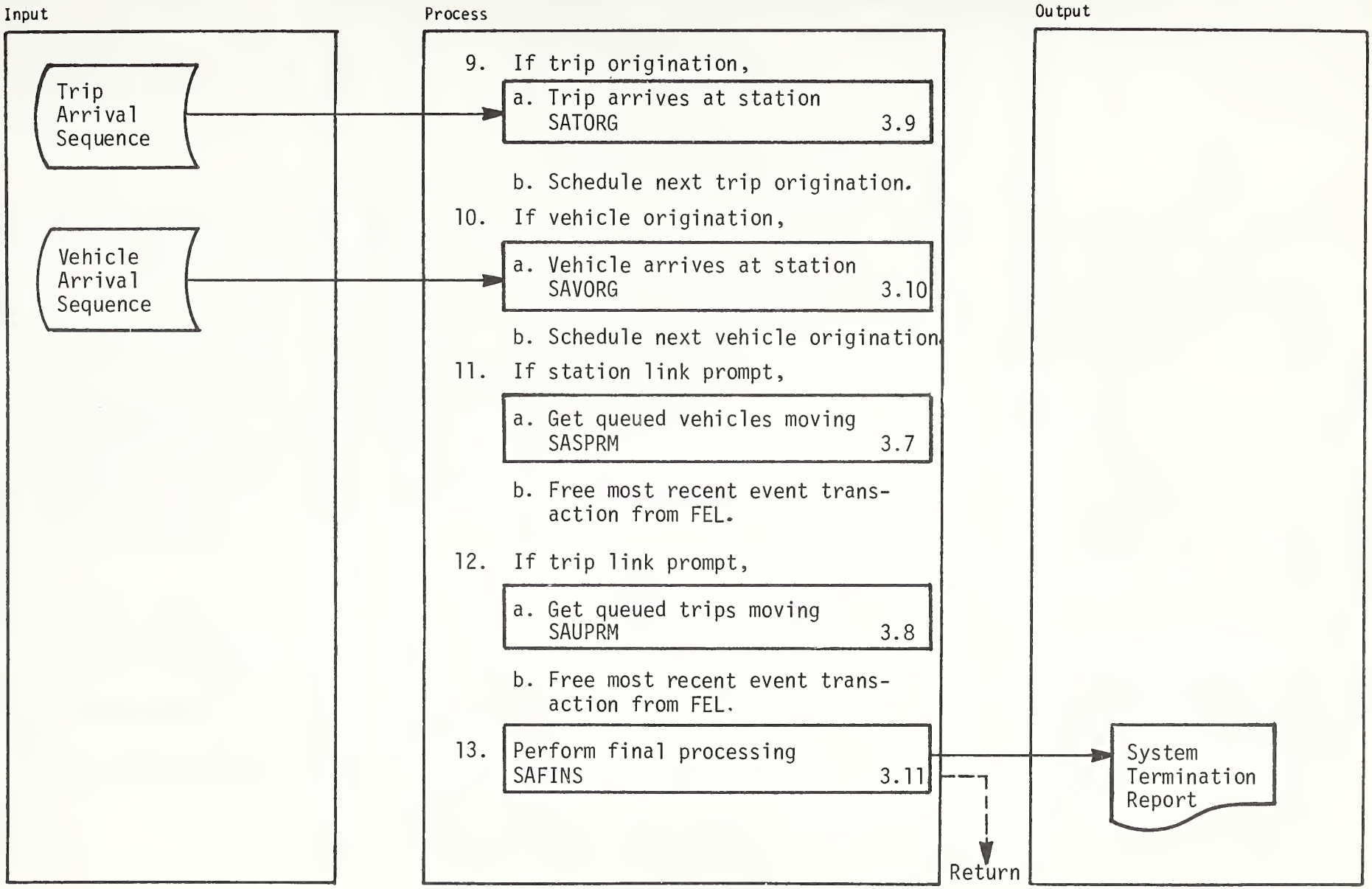
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 2

Diagram ID: 3.0 Name: SAMAIN (Section 2.2.2.1) Description: Model Processor Control



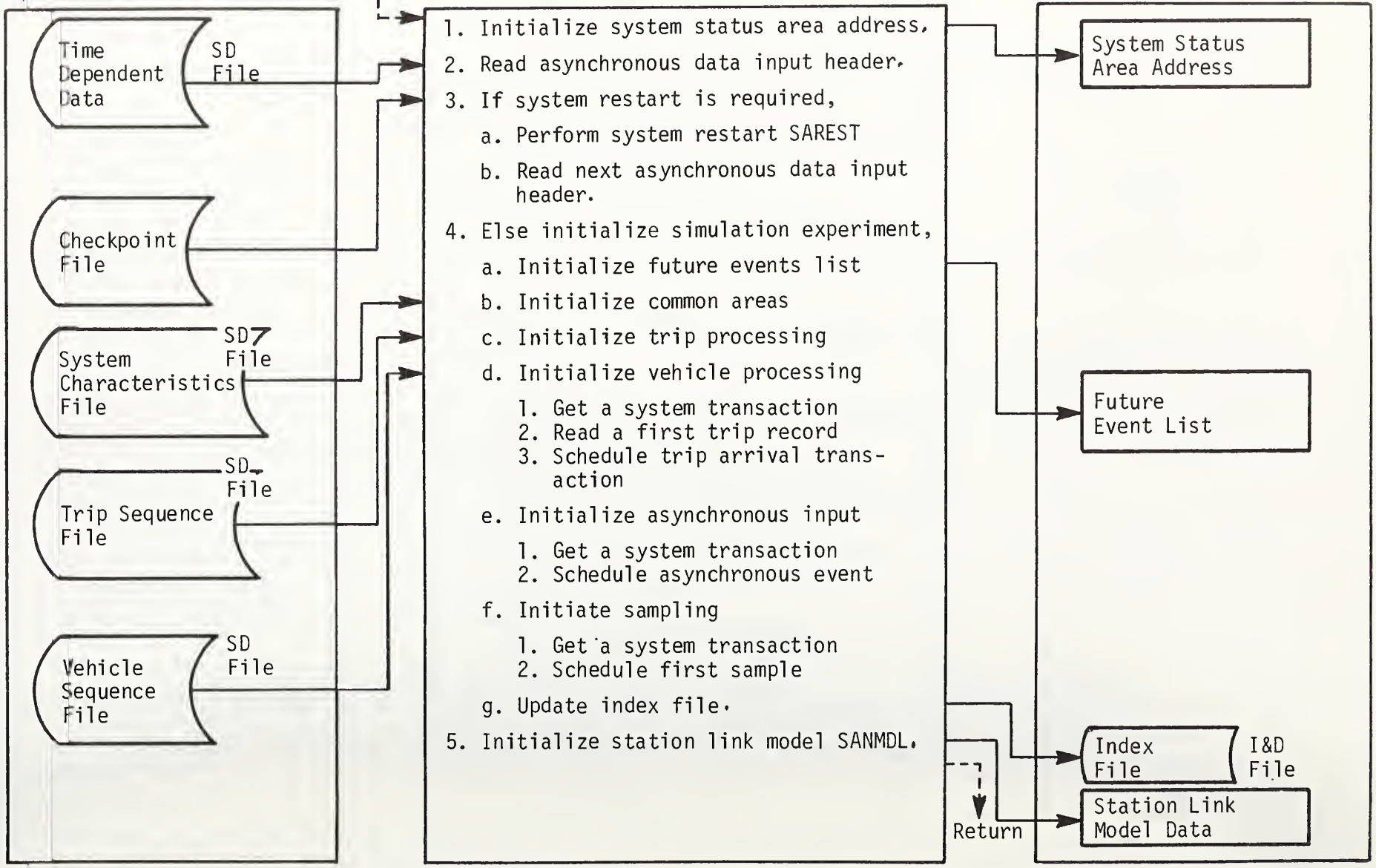
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 2 of 2

Diagram ID: 3.0 Name: SAMAIN Description: Model Processor Control



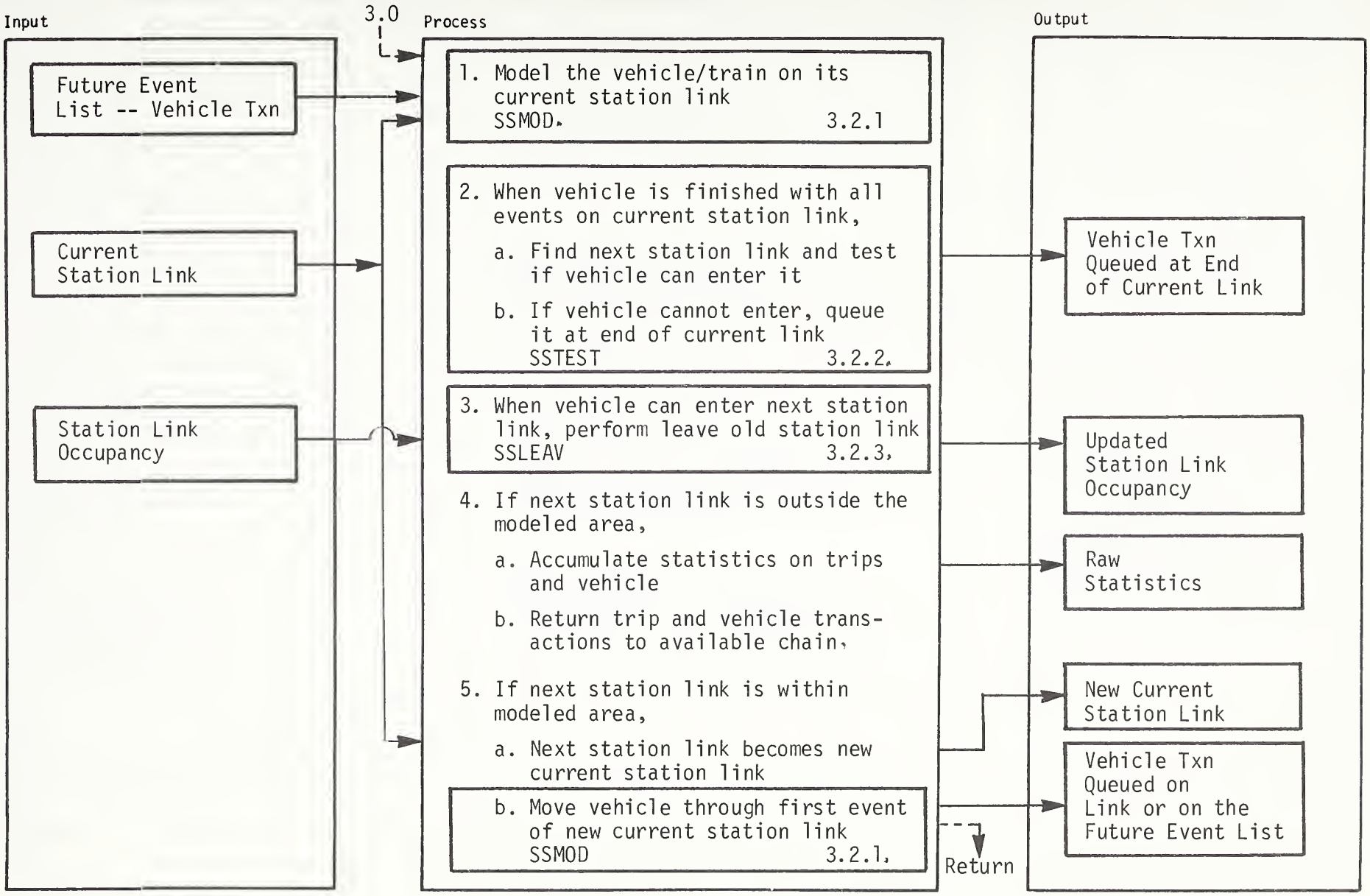
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.1 Name: SAINIT Description: System Initialization



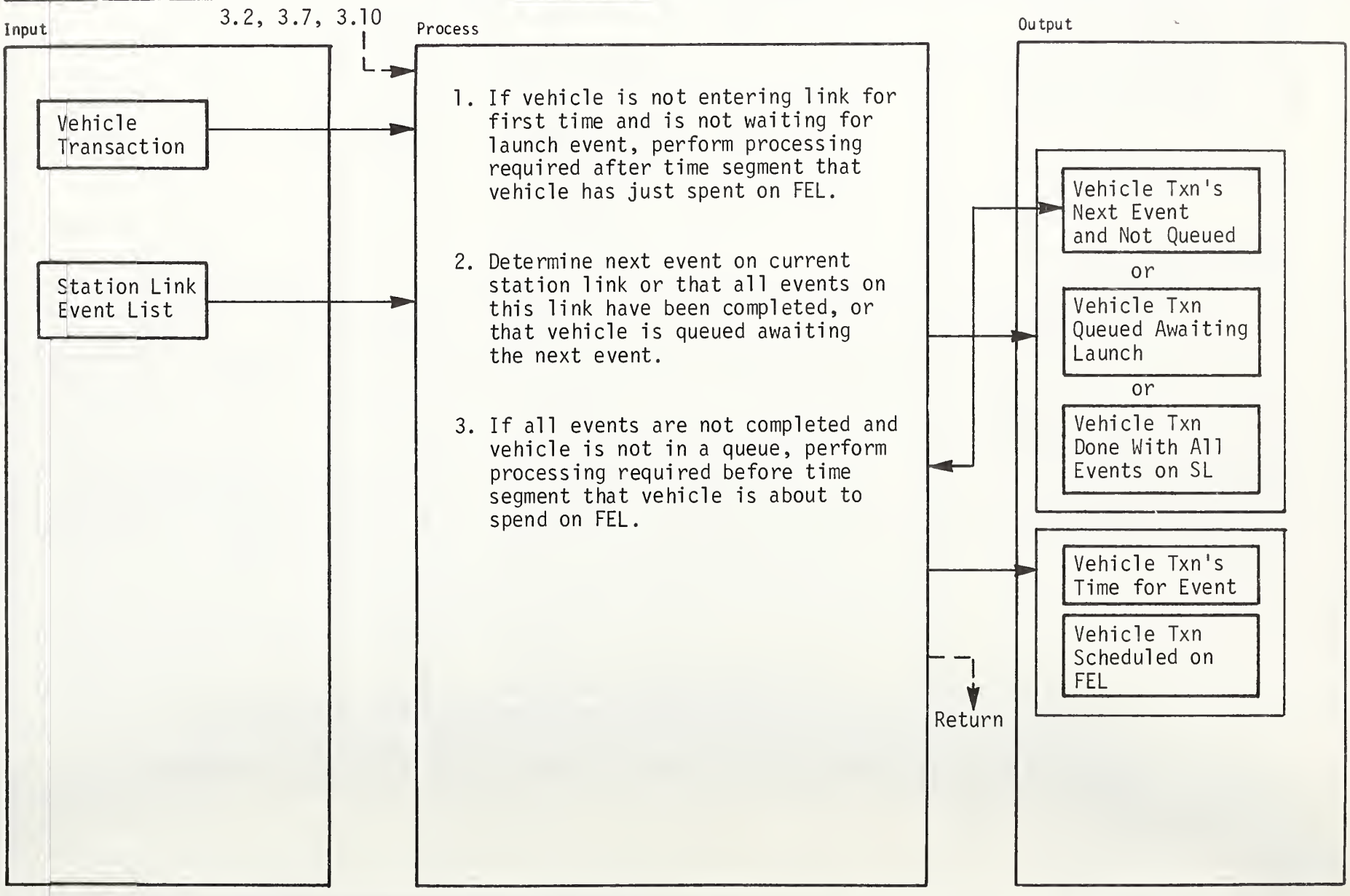
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.2 Name: SASCTL Description: Station Link Control



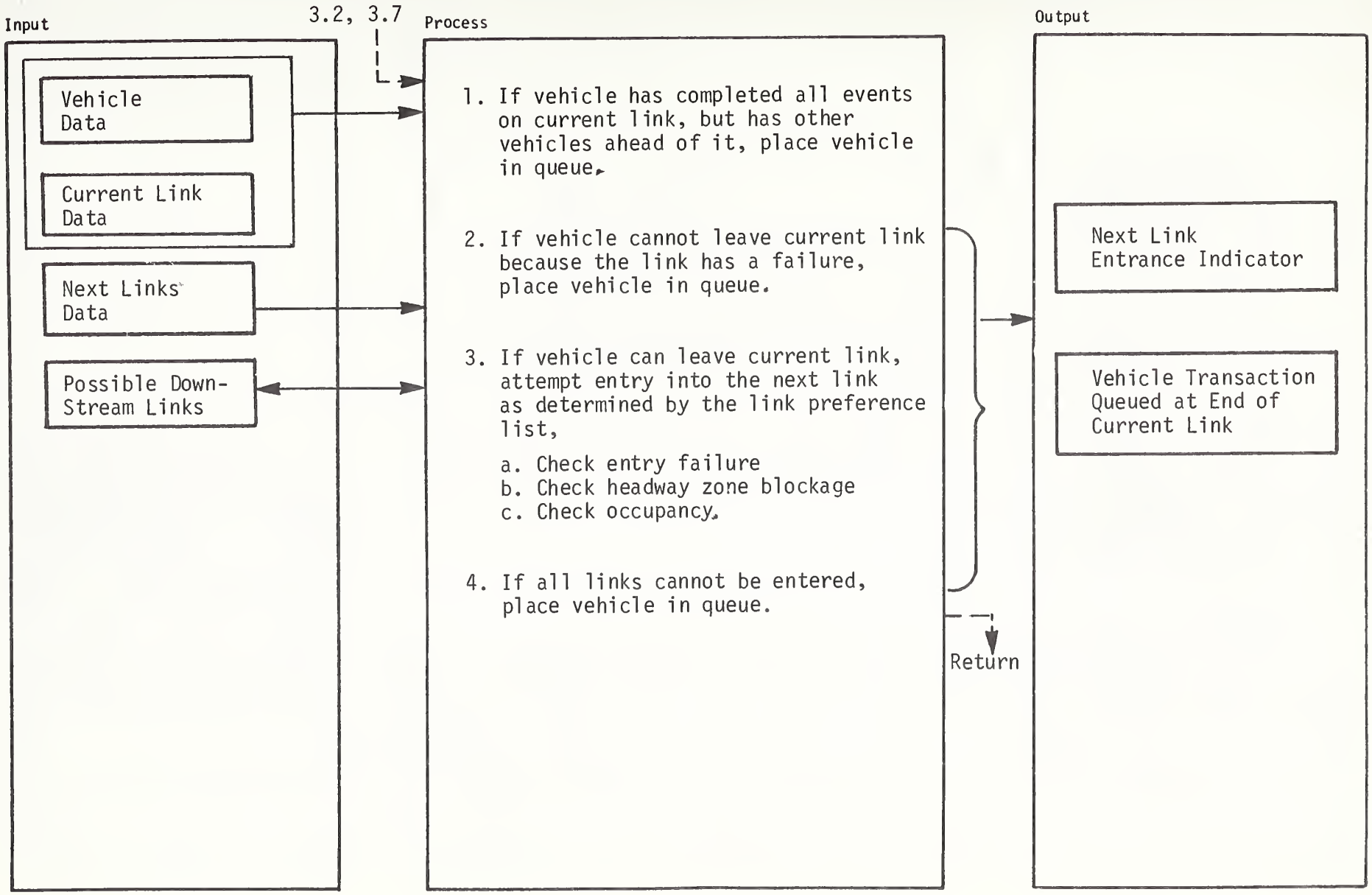
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: _____ of _____

Diagram ID: 3.2.1 Name: SSMOD Description: Station Link Model



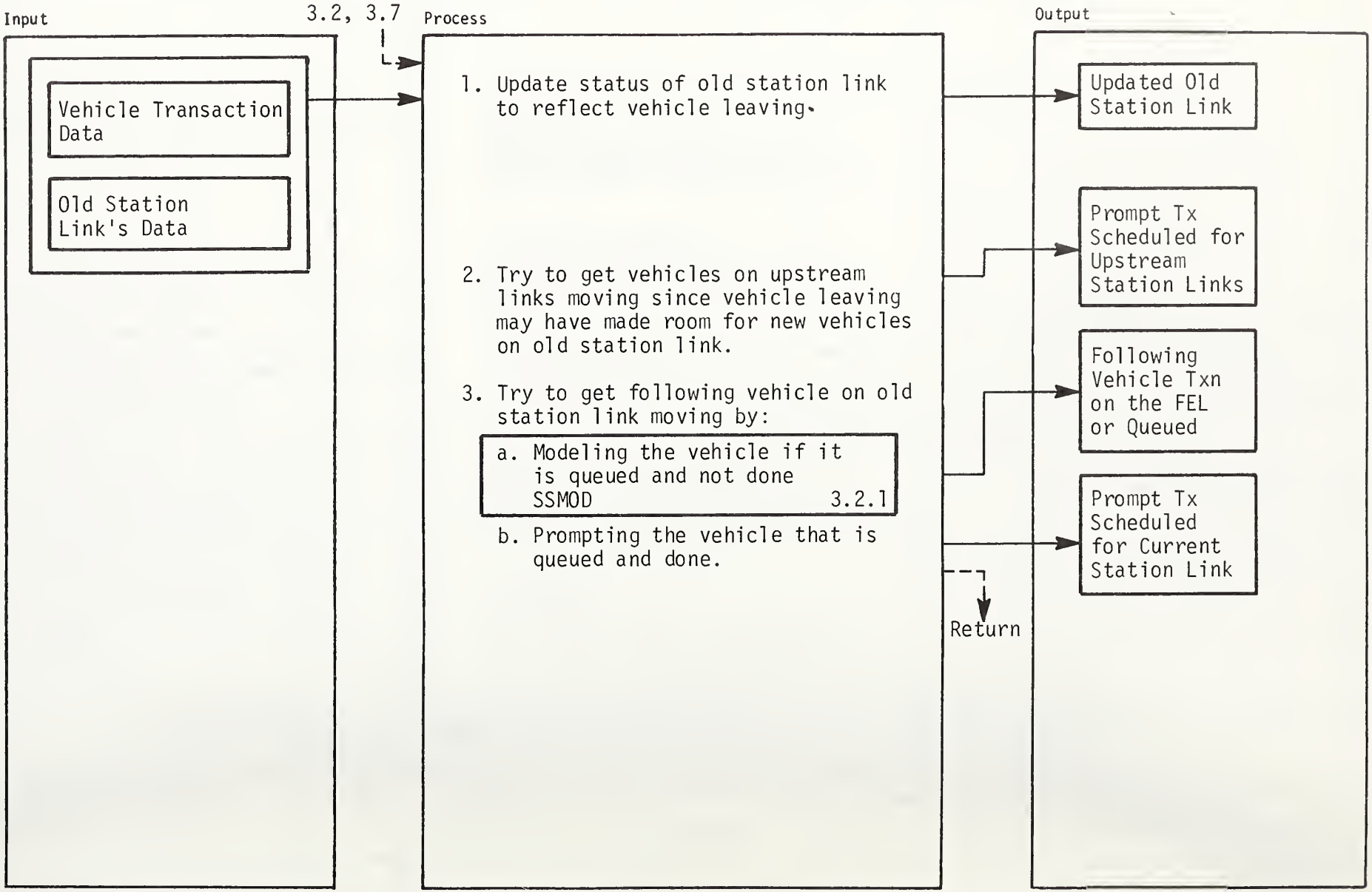
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.2.2 Name: SSTEST Description: Test Next Station Link



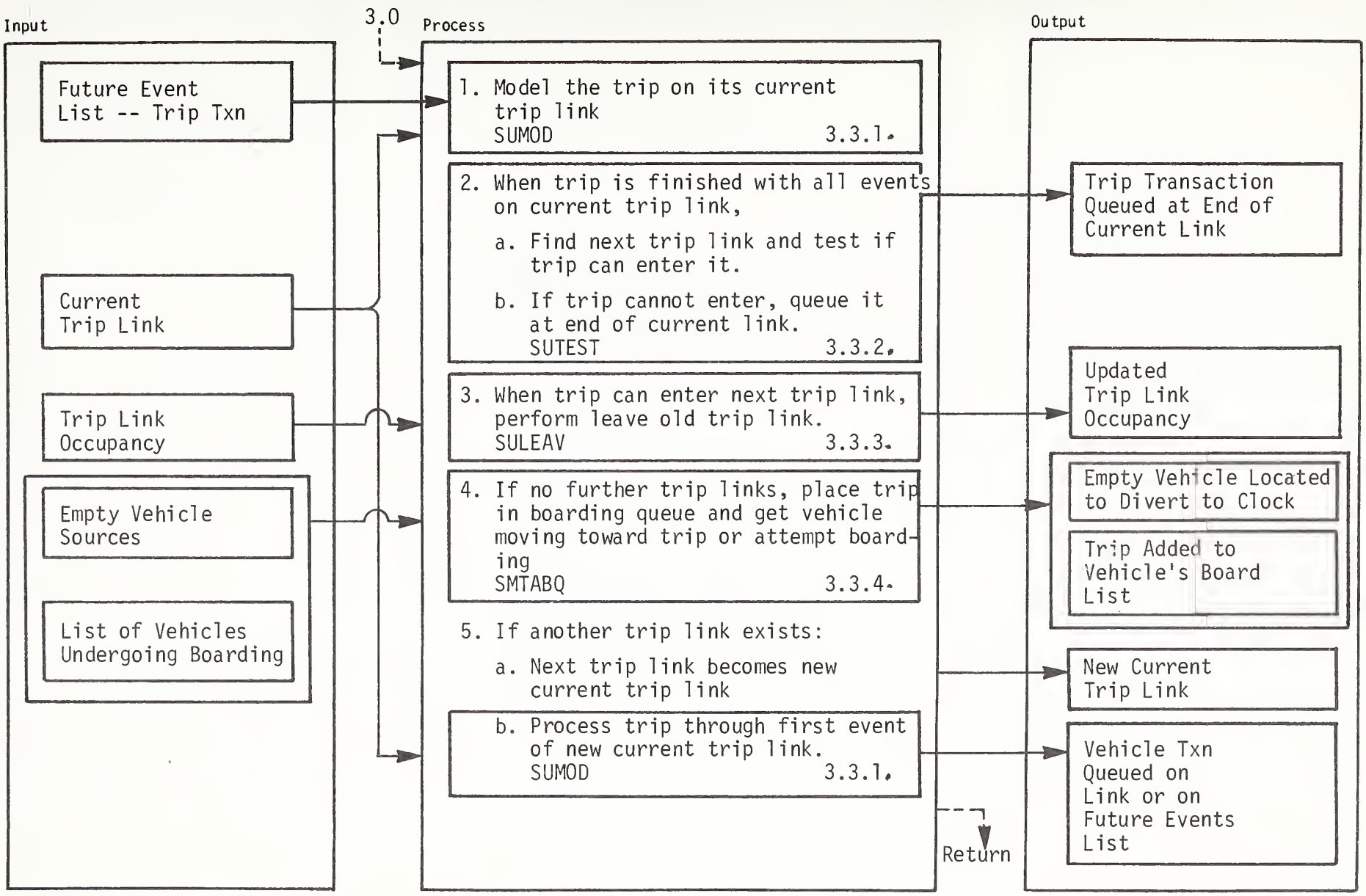
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.2.3 Name: SSLEAV Description: Leave Station Link



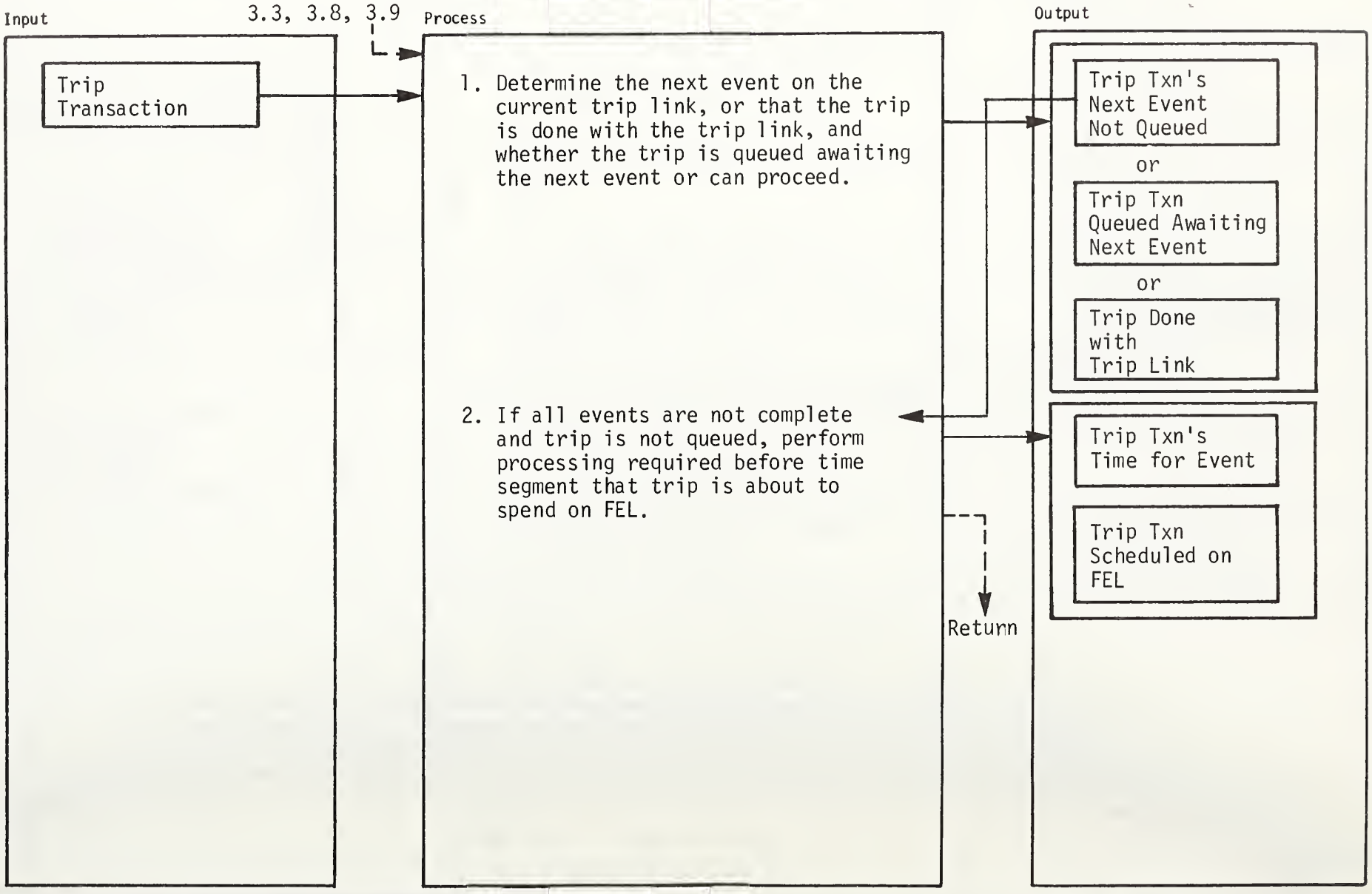
Author: _____ System/Program: AGT-SOS-DSM Date: _____ Page: 1 of 1

Diagram ID: 3.3 Name: SAUCTL Description: Trip Link Control



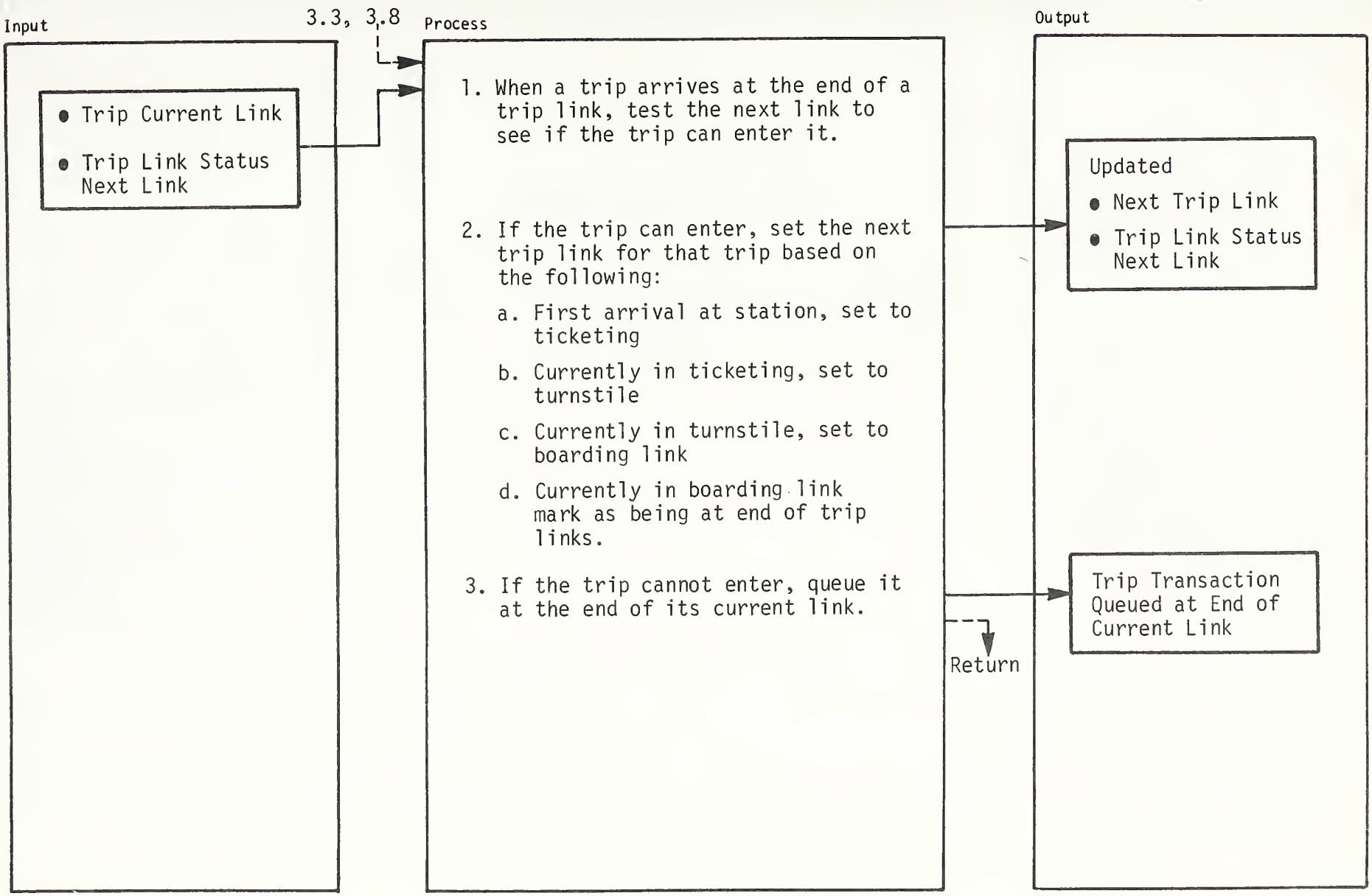
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.3.1 Name: SUMOD Description: Trip Link Model



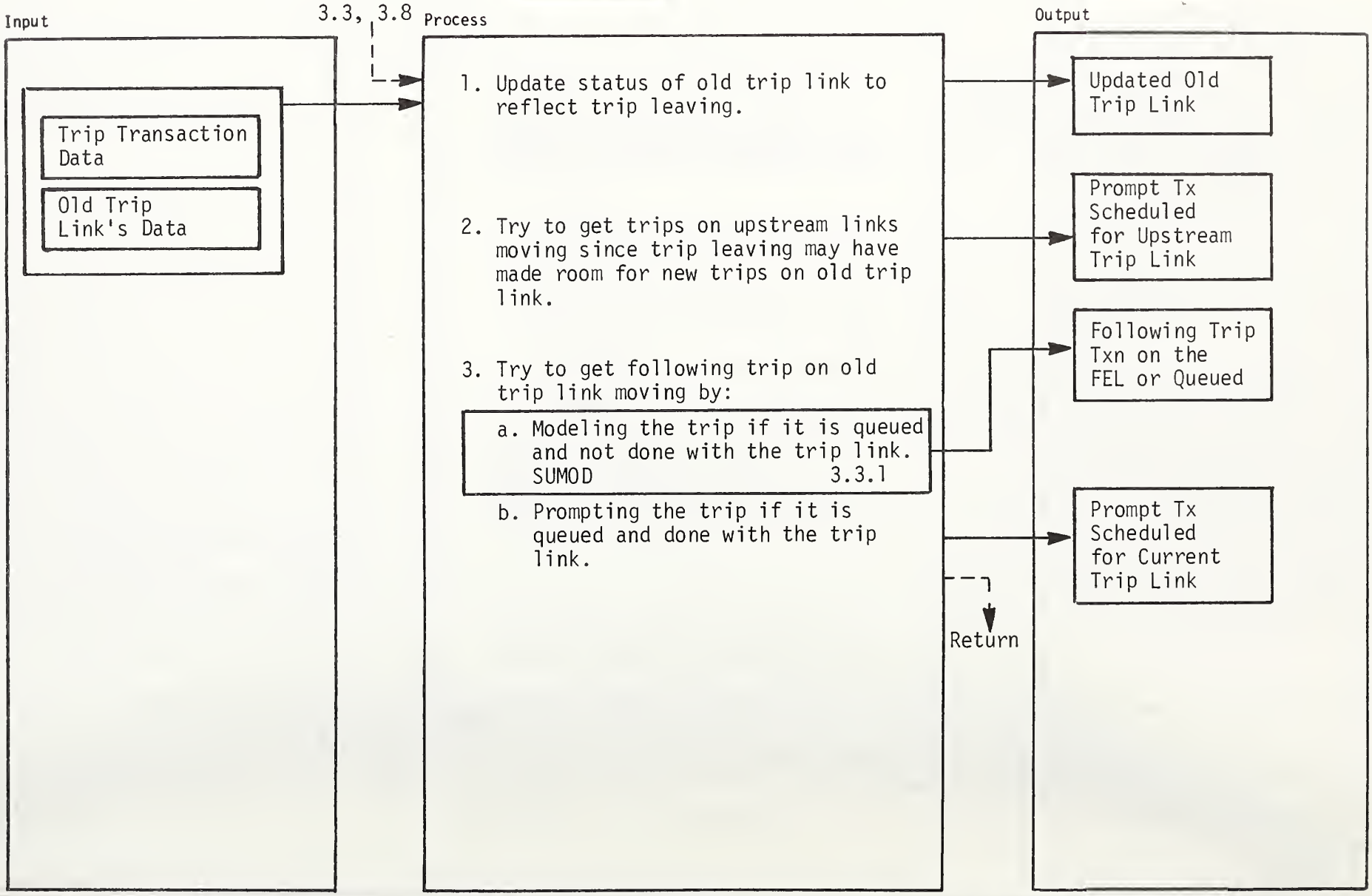
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.3.2 Name: SUTEST Description: Test Next Trip Link



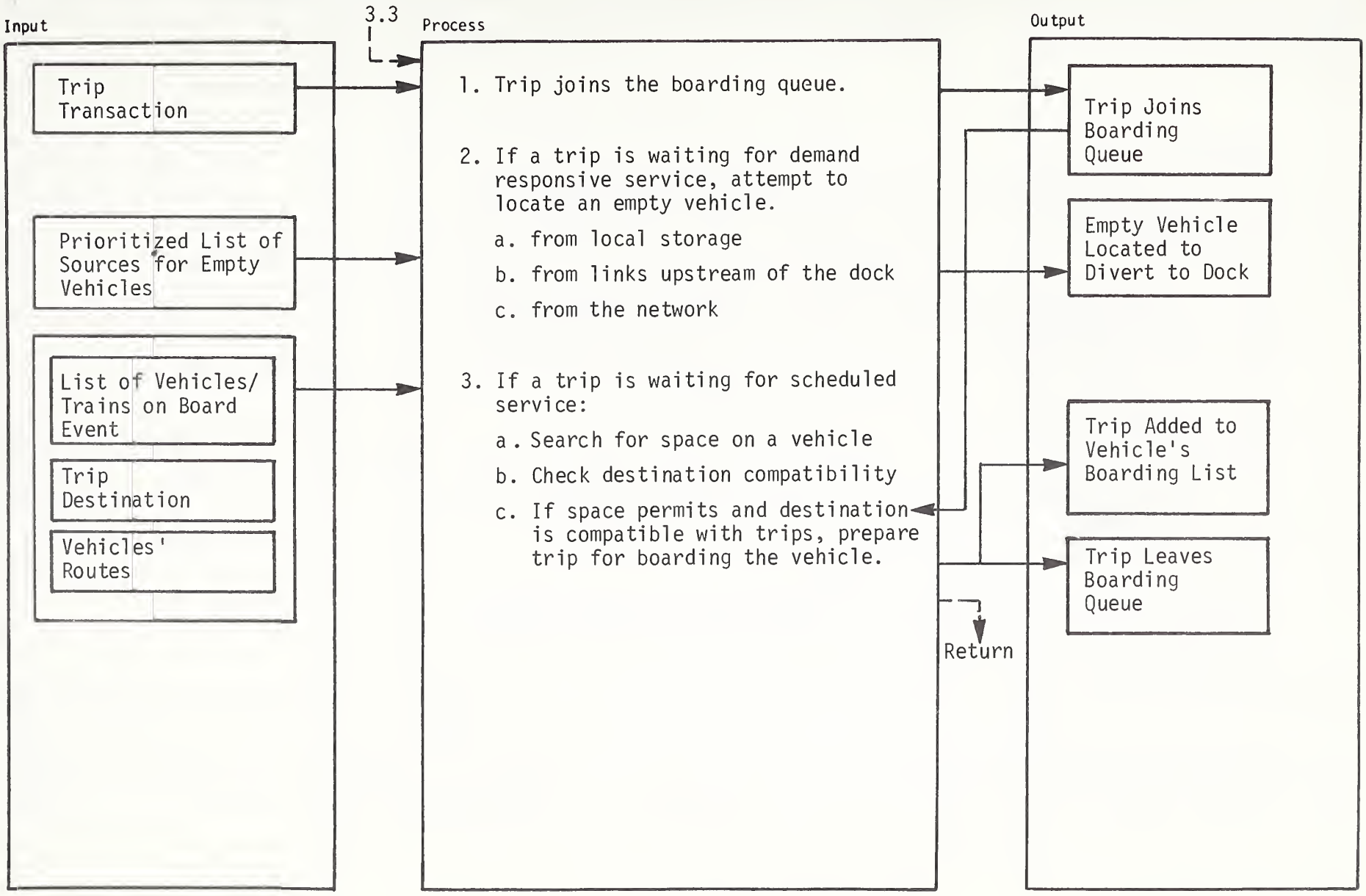
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.3.3 Name: SULEAV Description: Leave Trip Link



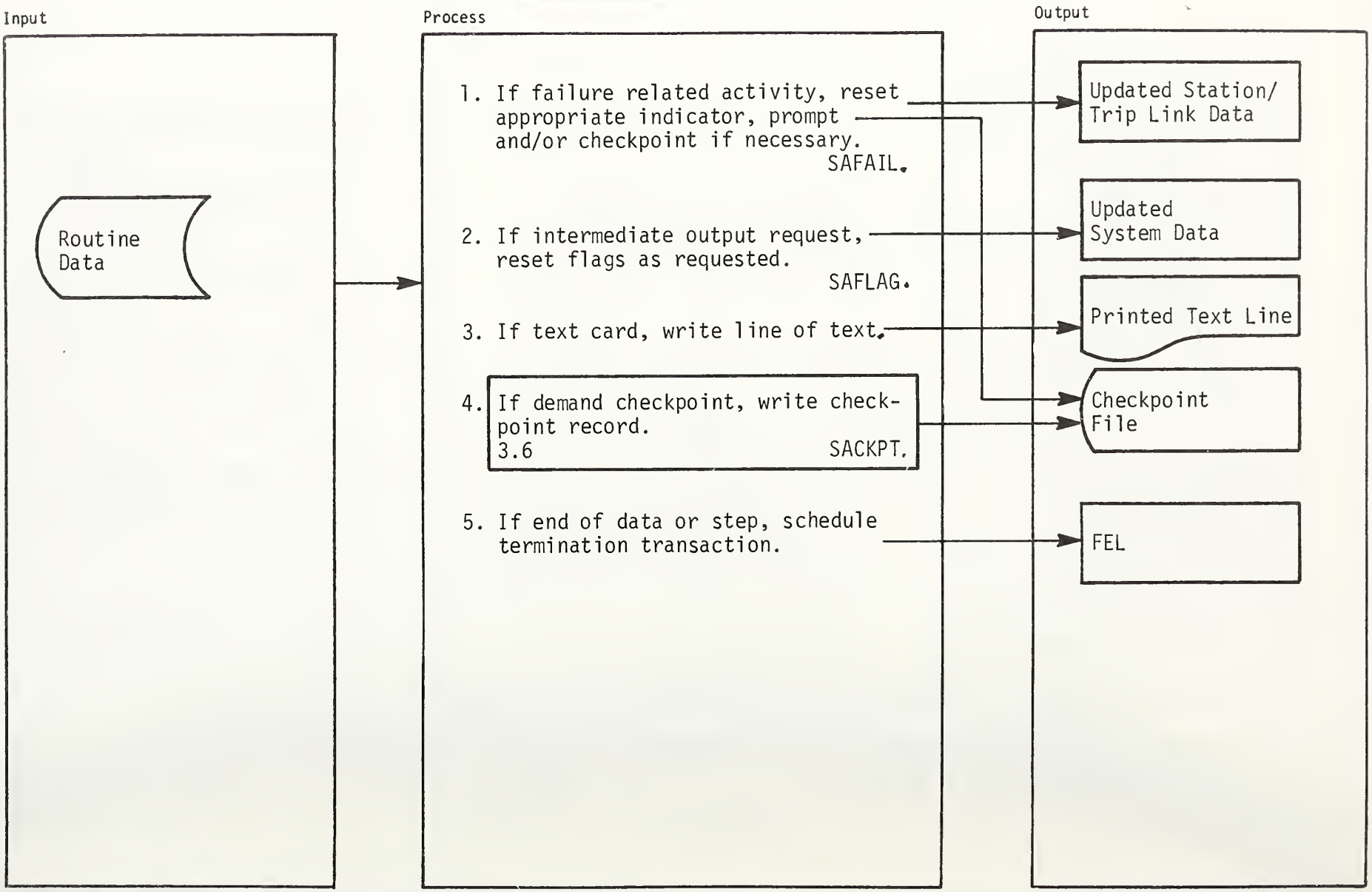
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.3.4 Name: SMTABQ Description: Trip at Boarding Queue



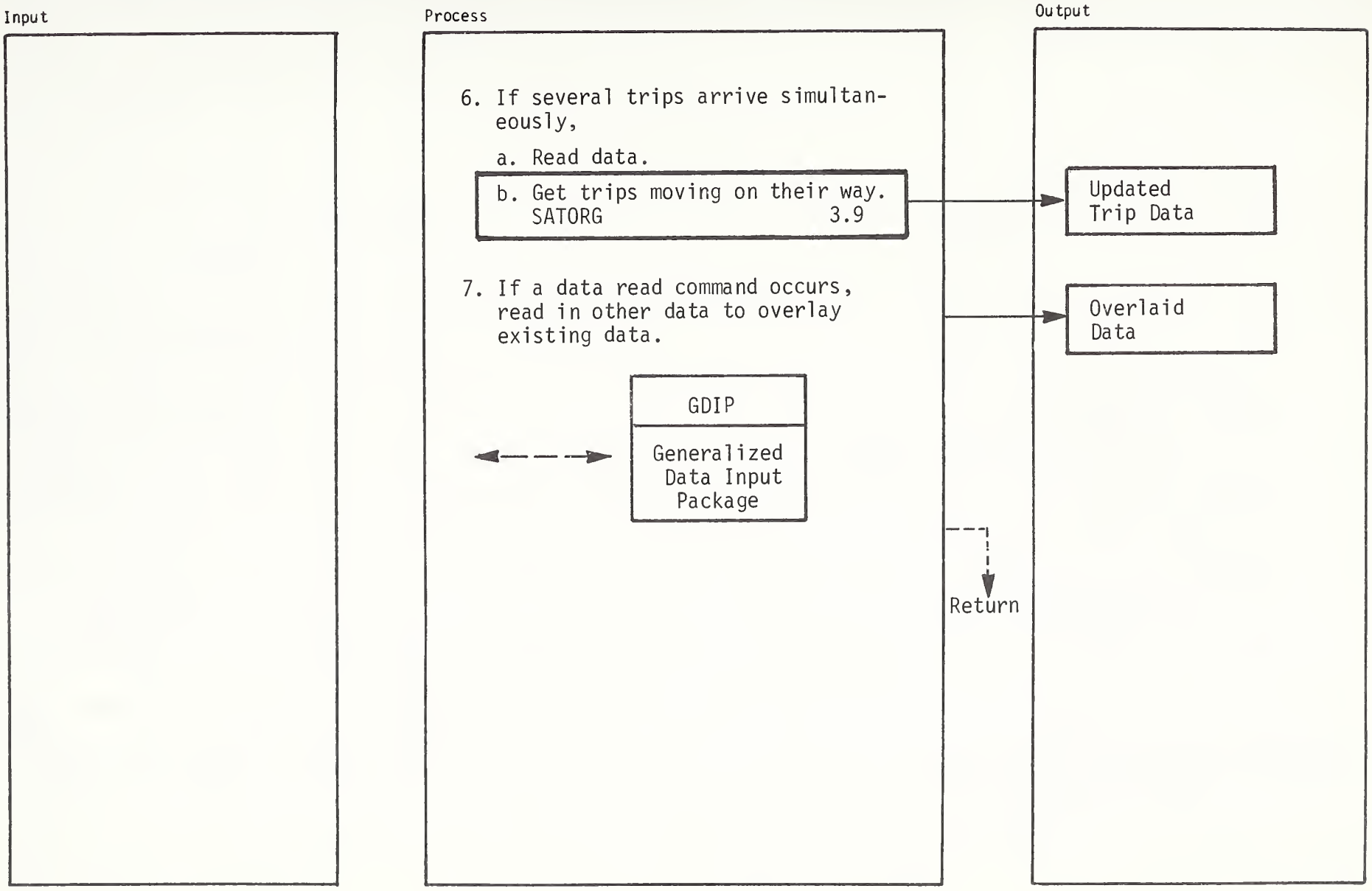
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 2

Diagram ID 3.4 Name: SAASYN Description: Process Asynchronous Command



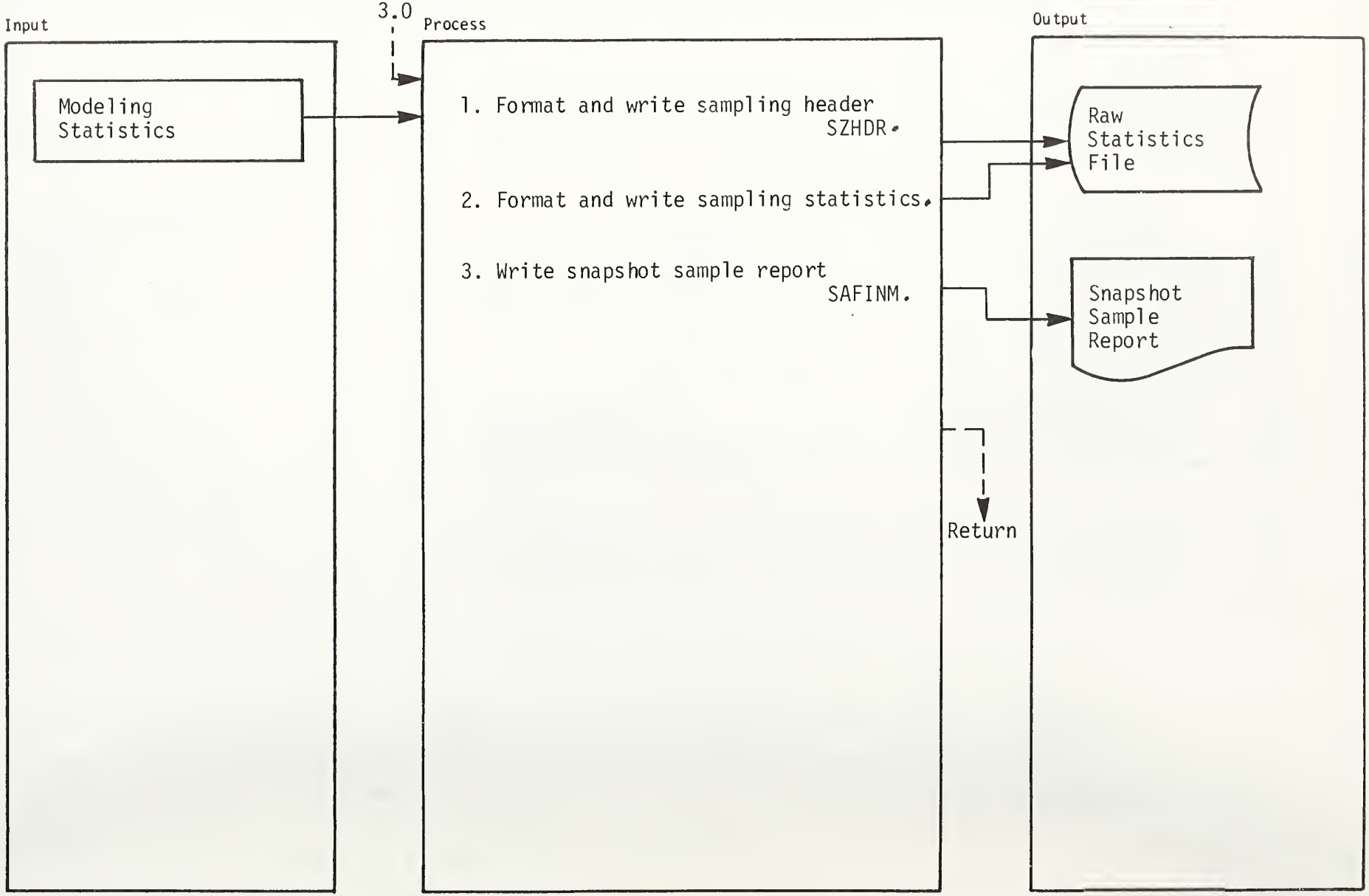
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 2 of 2

Diagram ID 3.4 Name: SAASYN Description: Process Asynchronous Commands



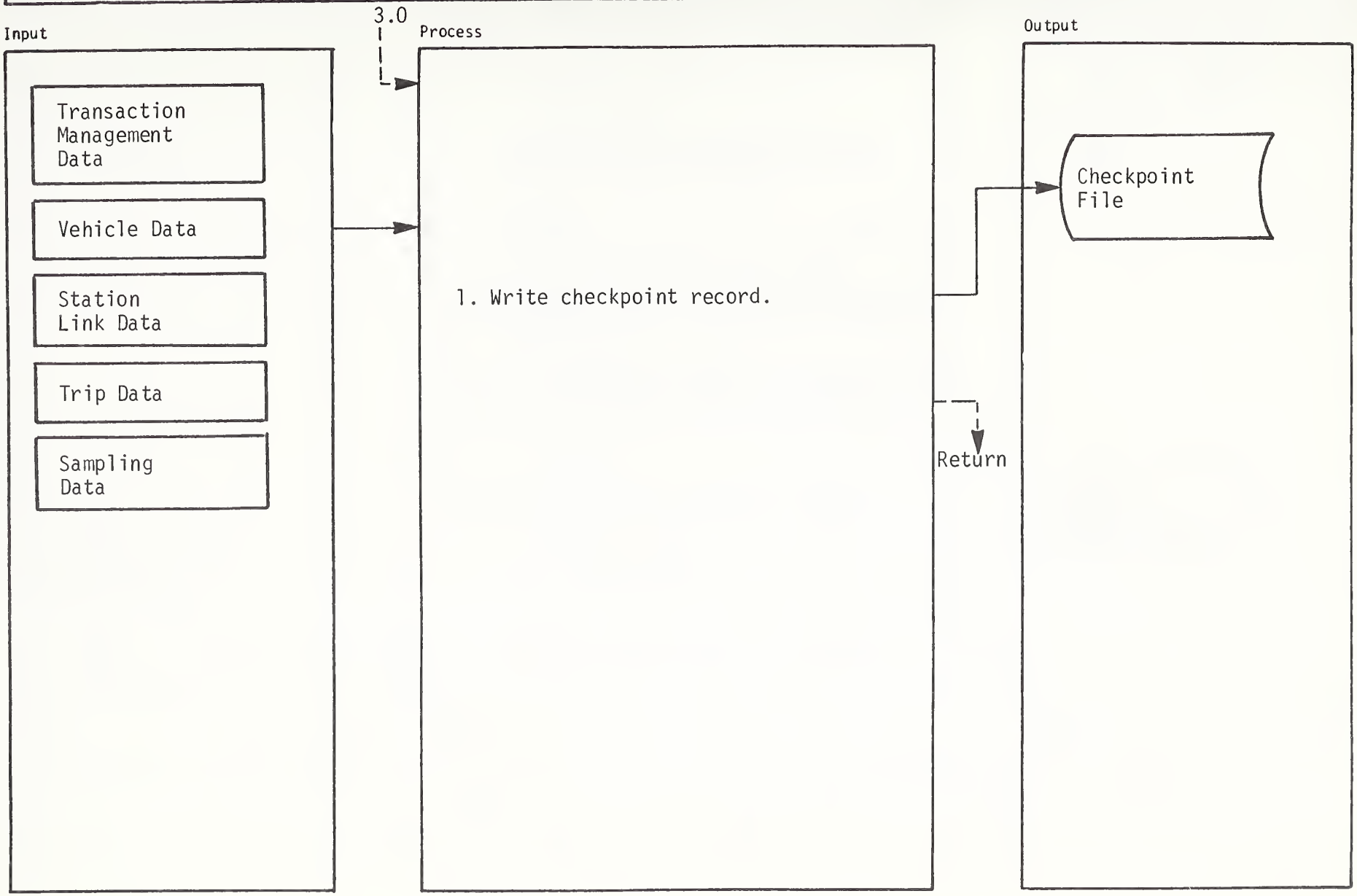
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.5 Name: SASAMP Description: Perform Output Sample



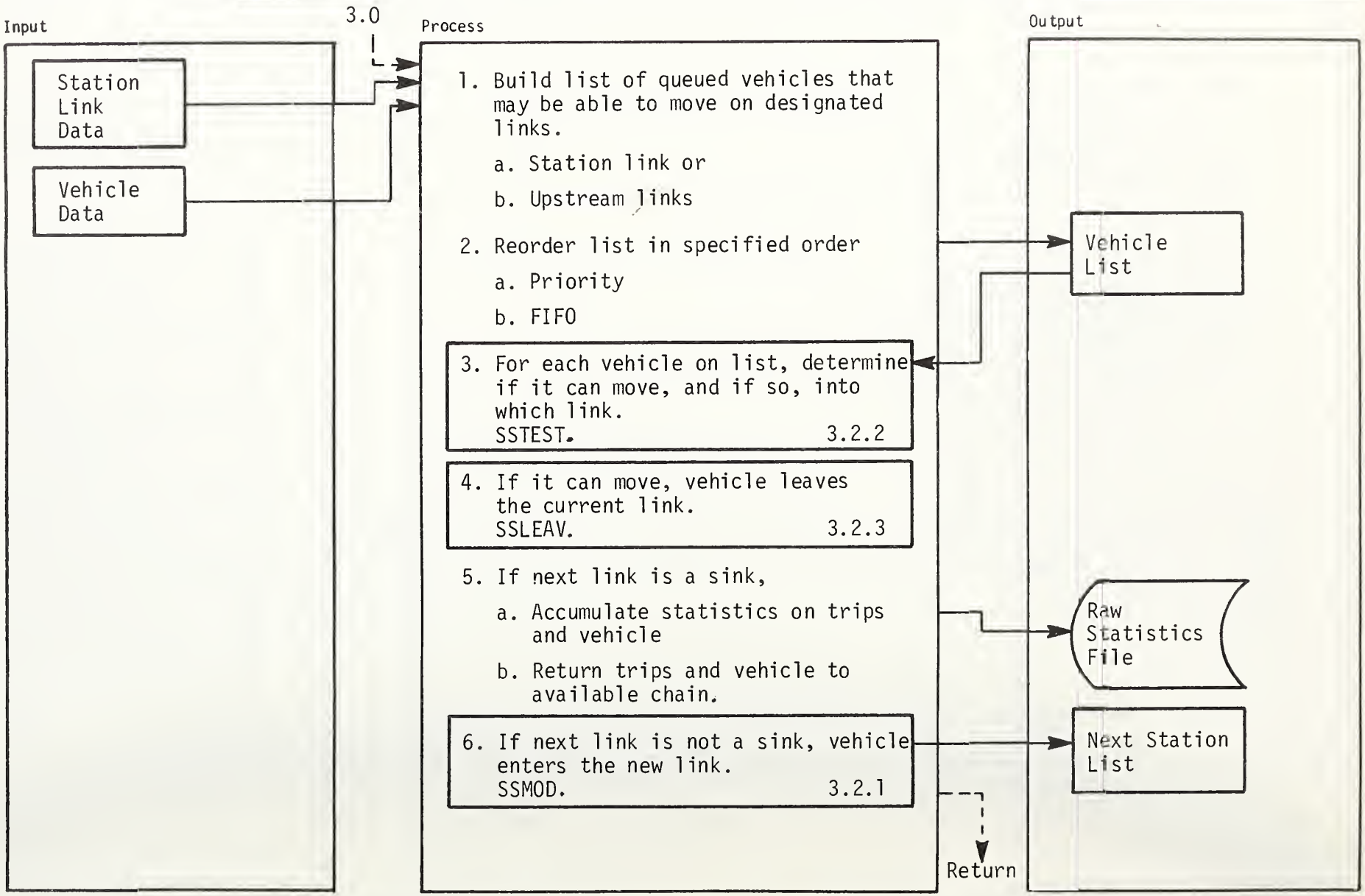
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.6 Name: SACKPT Description: Perform System Checkpointing



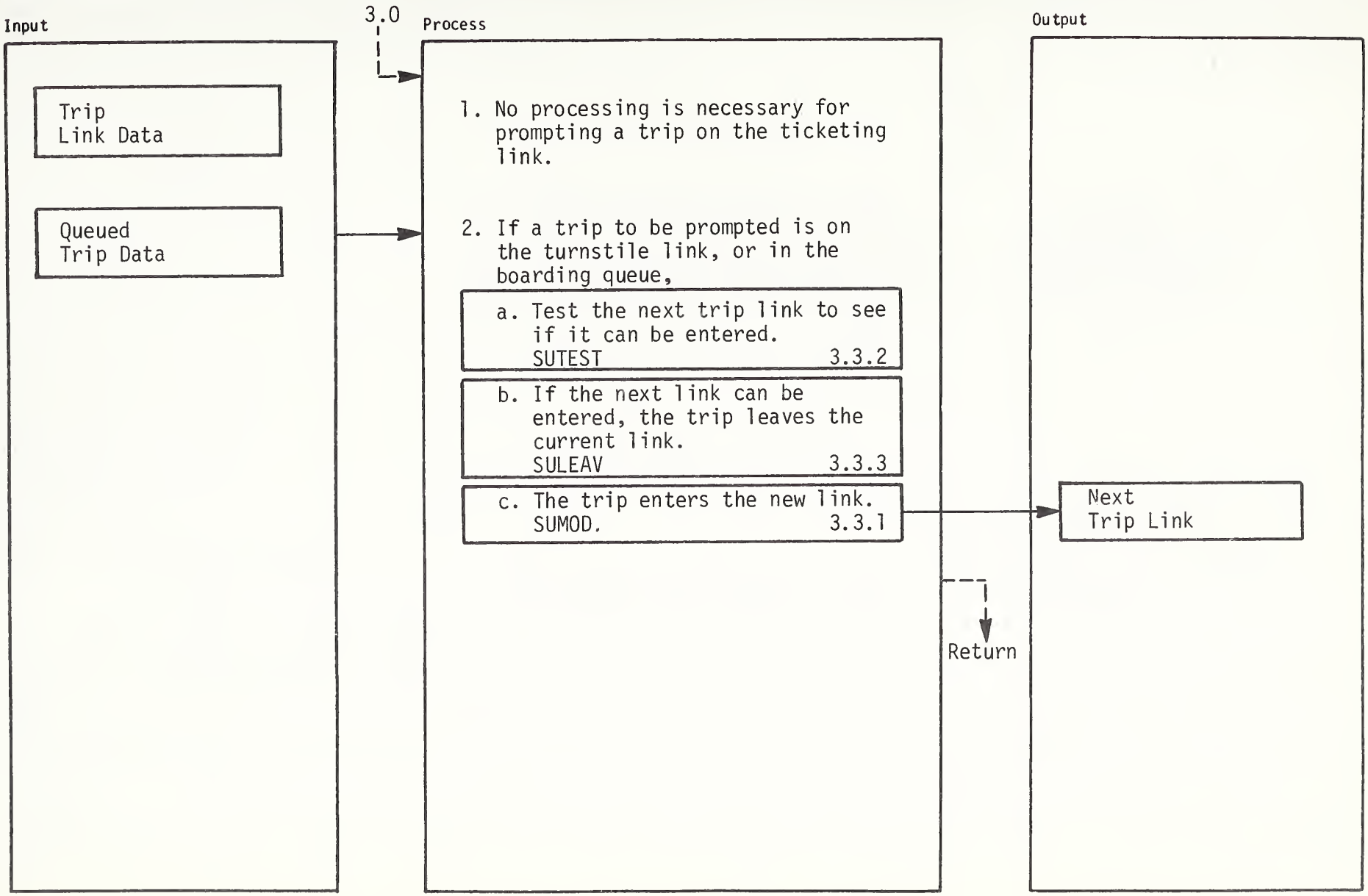
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.7 Name: SASPRM Description: Station Link Prompt



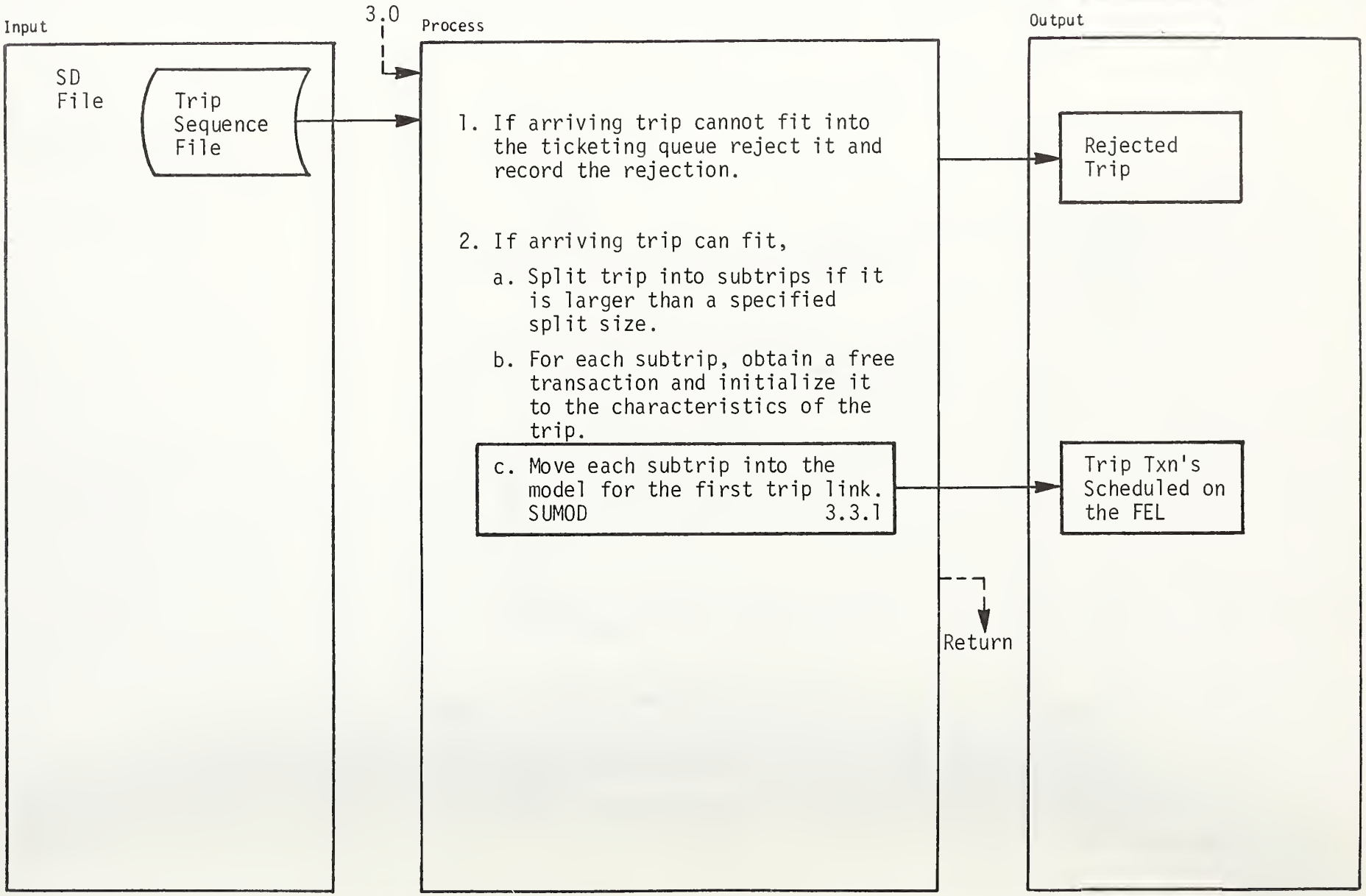
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.8 Name: SAUPRM Description: Trip Link Prompt



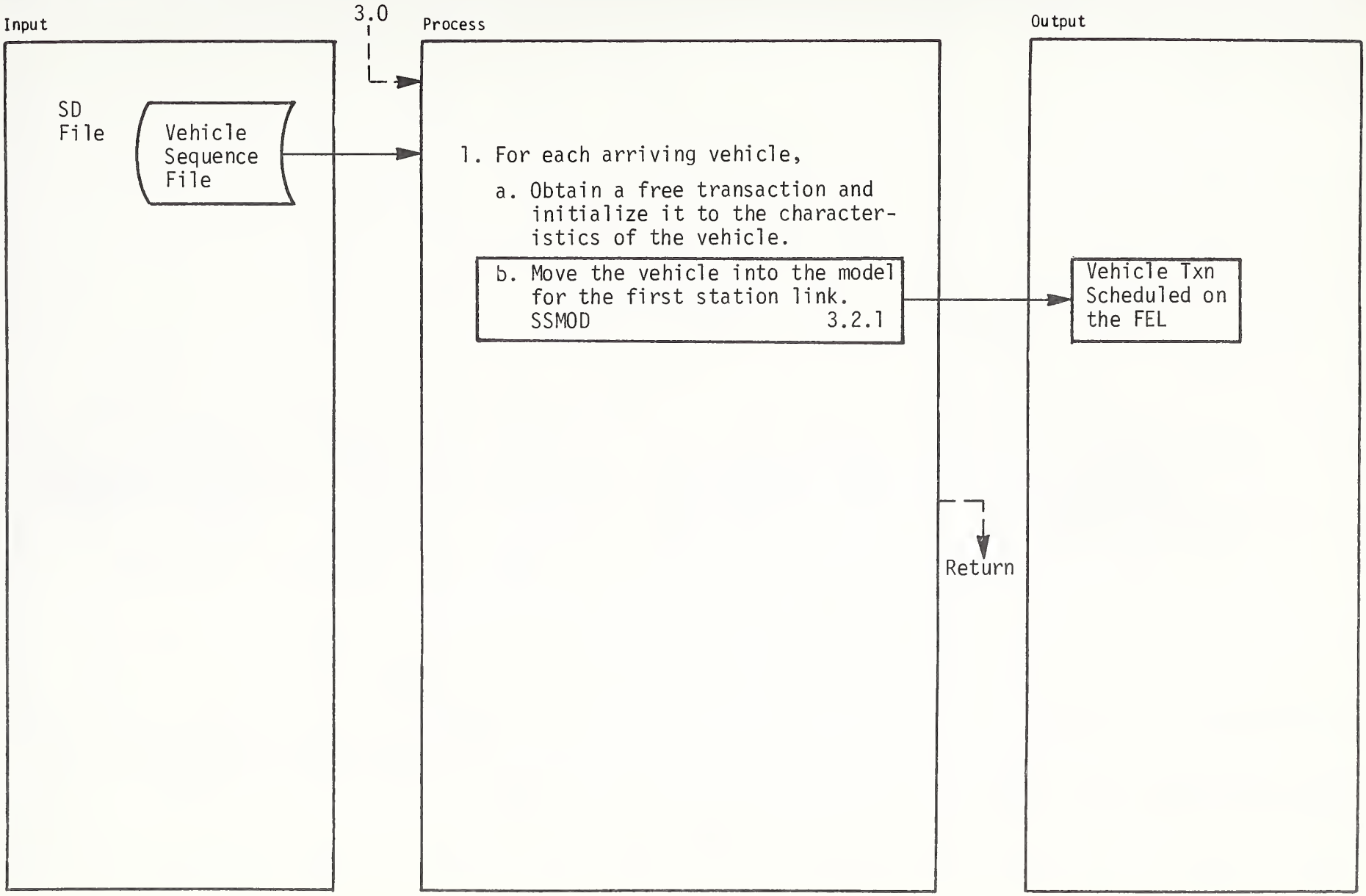
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID 3.9 Name: SATORG Description: Move Arriving Trip



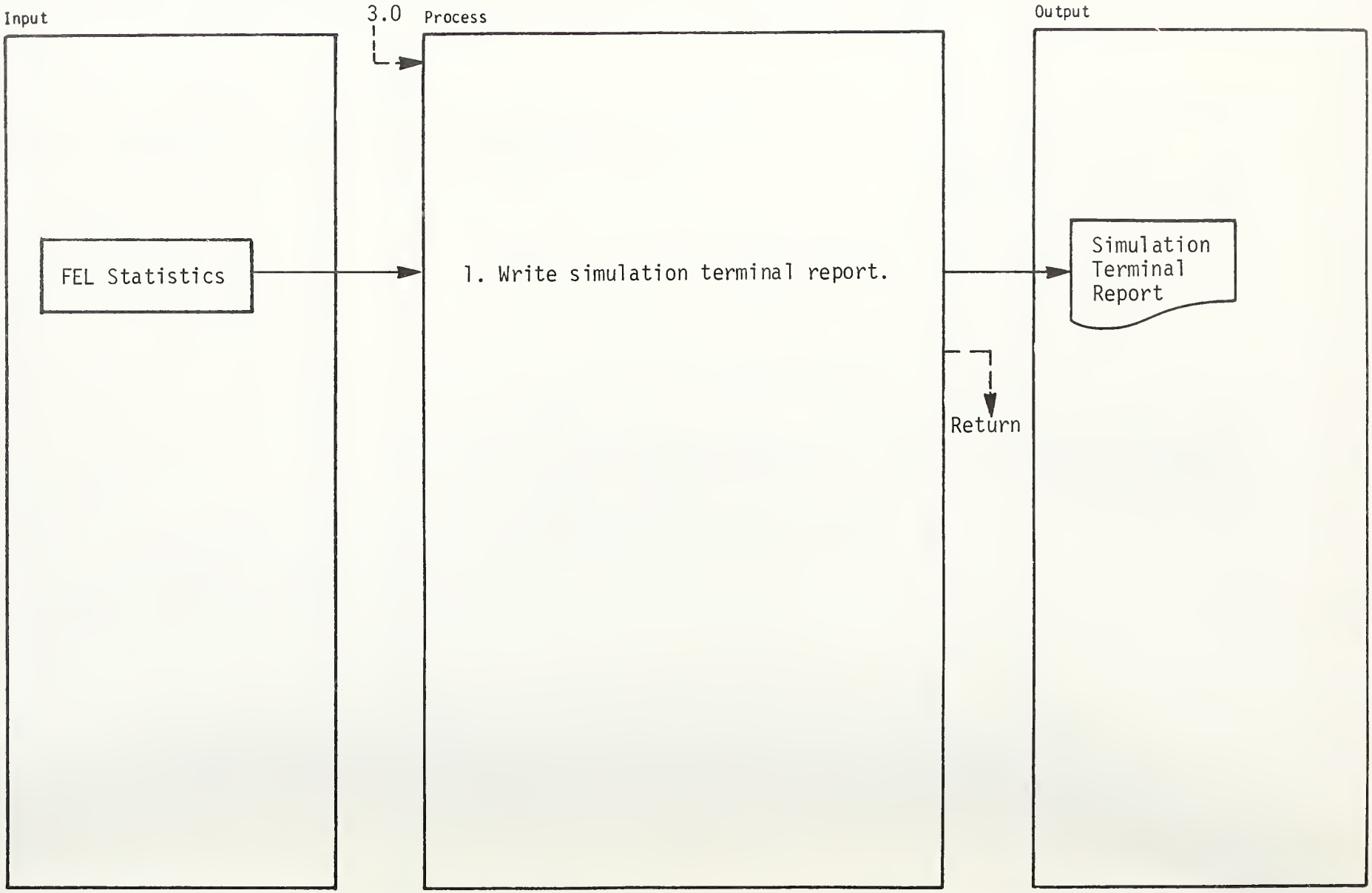
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID 3.10 Name: SAVORG Description: Move Arriving Vehicle



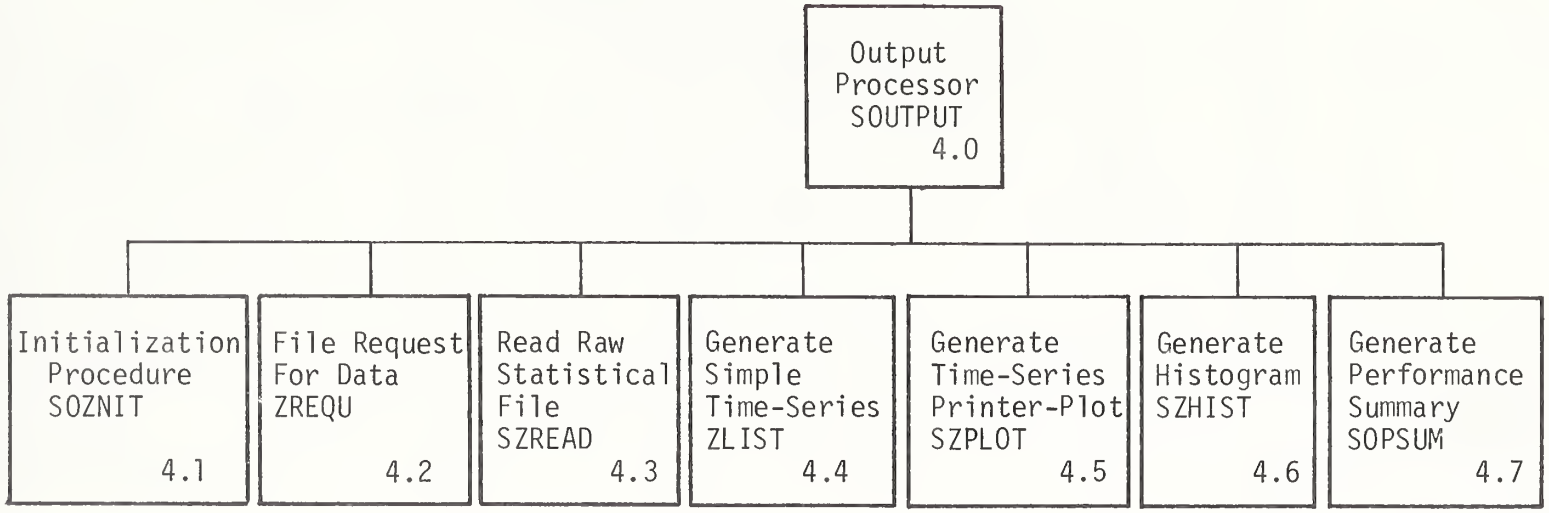
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 3.11 Name: SAFINS Description: Simulation Termination Processing



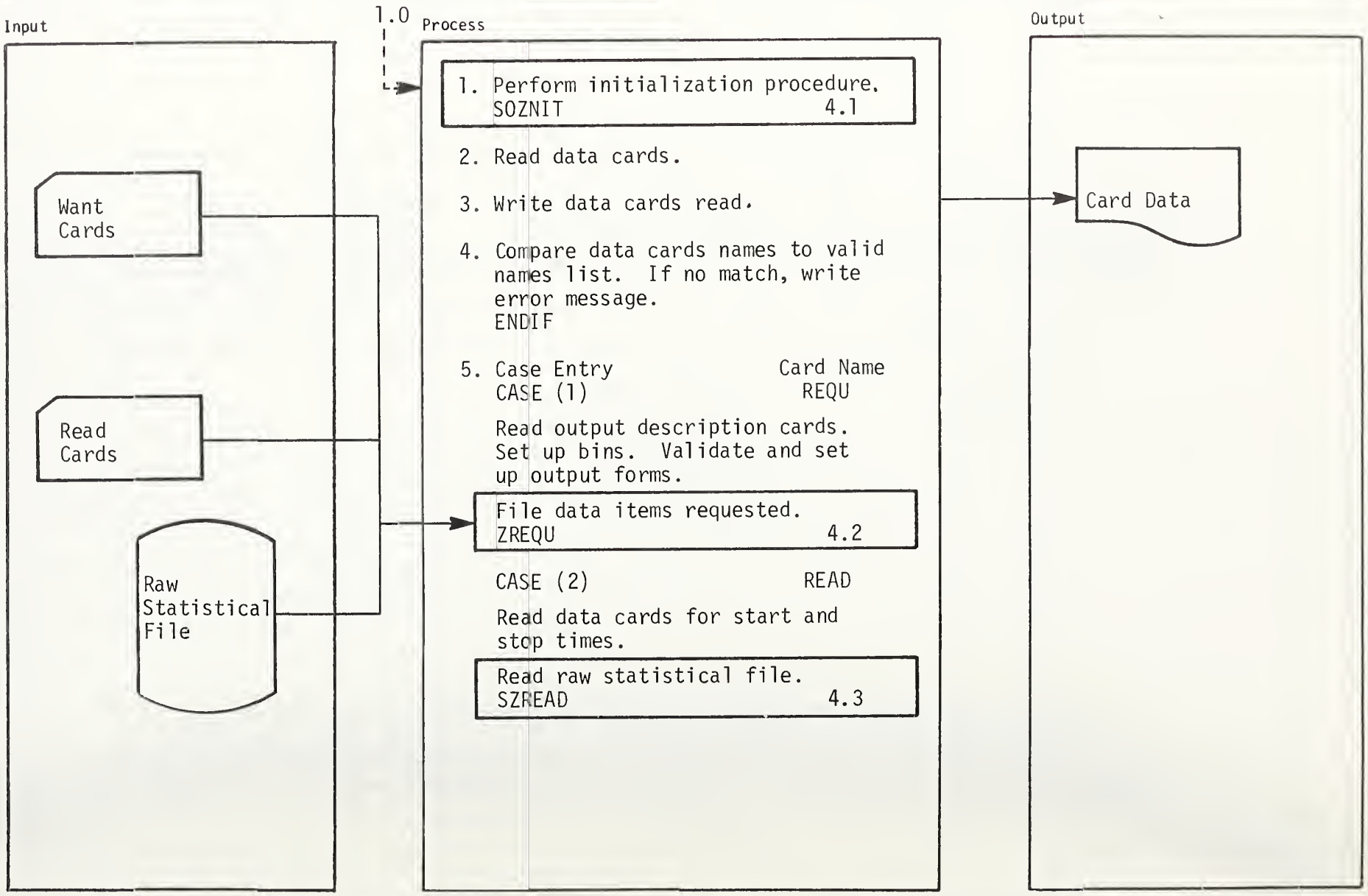
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: VTOC Name: Visual Table of Contents Description: Output Processor



Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 2

Diagram ID: 4.0 Name: SOUTPT Description: Output Processor for DSM



Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 2 of 2

Diagram ID: 4.0 Name: SOUTPT Description: Output Processor for DSM

Input

Process

Process requested data.

Case Entry	Input Forms
CASE (3,4)	LIST
Generate sample time-series or summary ZLIST	4.4
CASE (2)	PLOT
Generate time-series printer-plot. SZPLOT	4.5
CASE (1)	HIST
Generate class-interval frequency distribution histogram. ZHIST	4.6
CASE (7)	PERF
Generate performance summary SOPSUM	4.7

ENDCASE

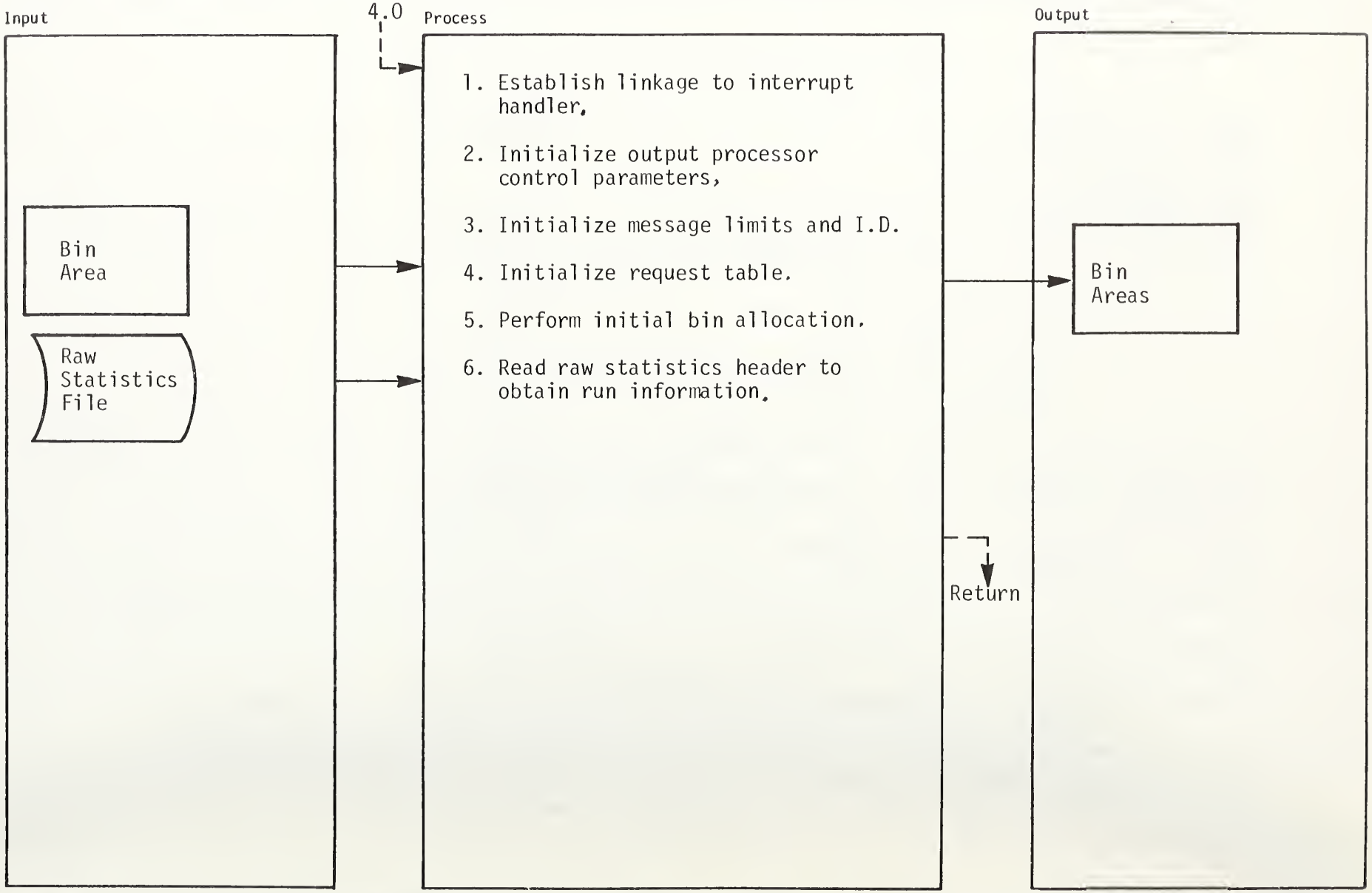
6. If no error, then terminate normally.
   Else, error message.
   ENDIF

Output

- List
- Plot
- Histogram
- Statistical Summary
- Errors
- Performance Summary File

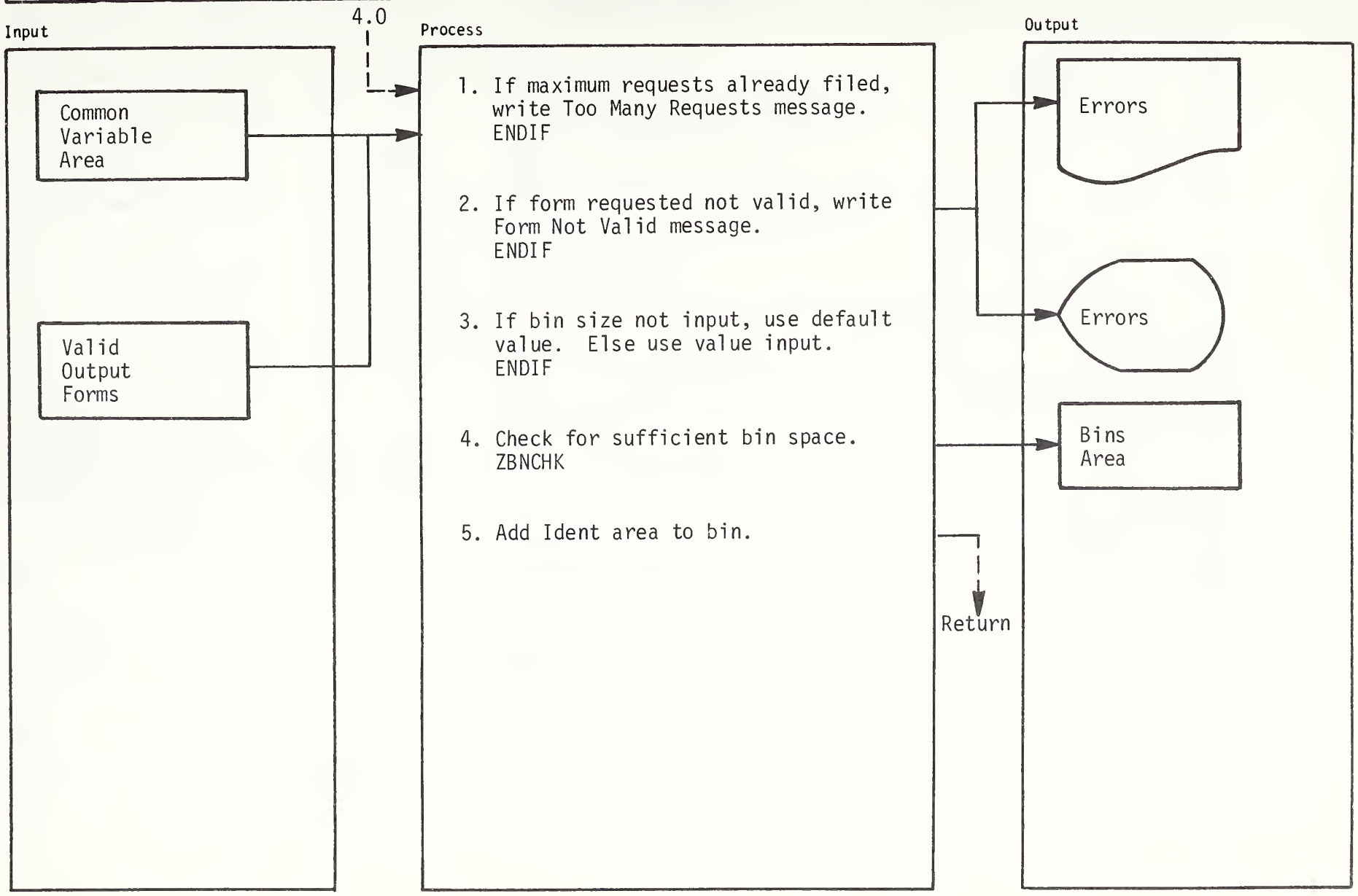
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 4.1 Name: SOZNIT Description: Initialization Procedure



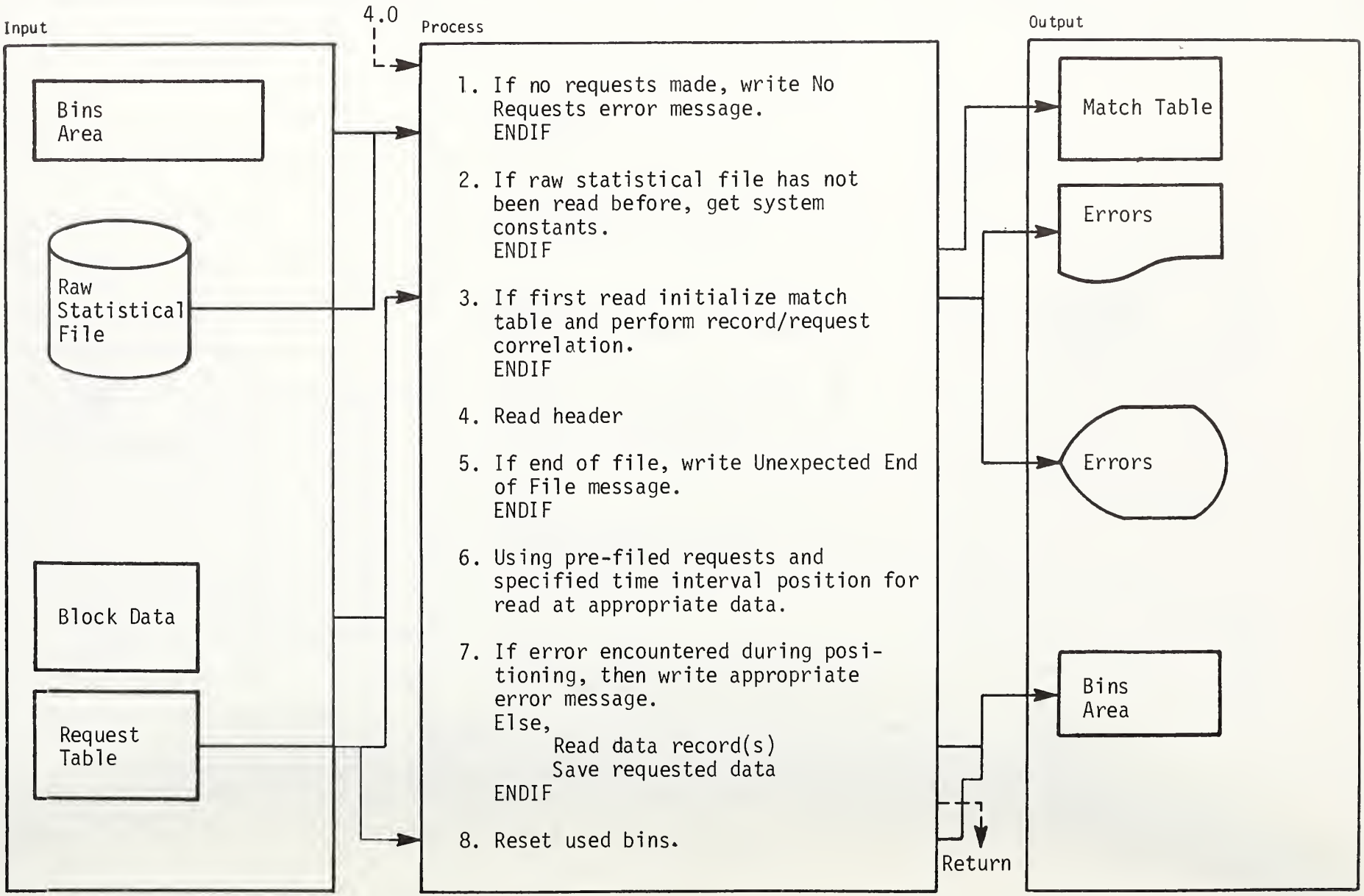
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 4.2 Name: ZREQU Description: File a Request for Data



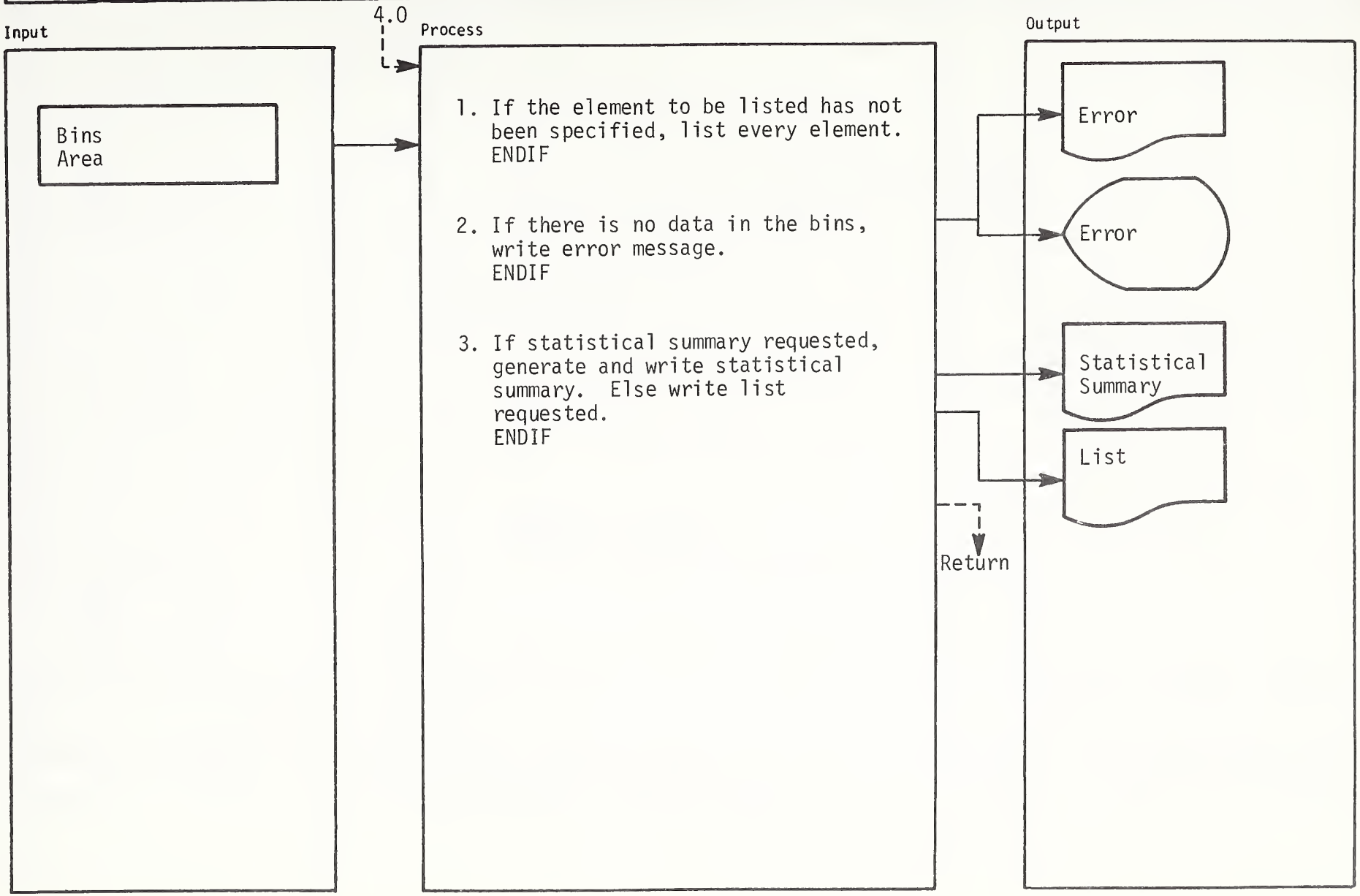
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 4.3 Name: SZREAD (Section 2.3.2.4) Description: Read Raw Statistical File



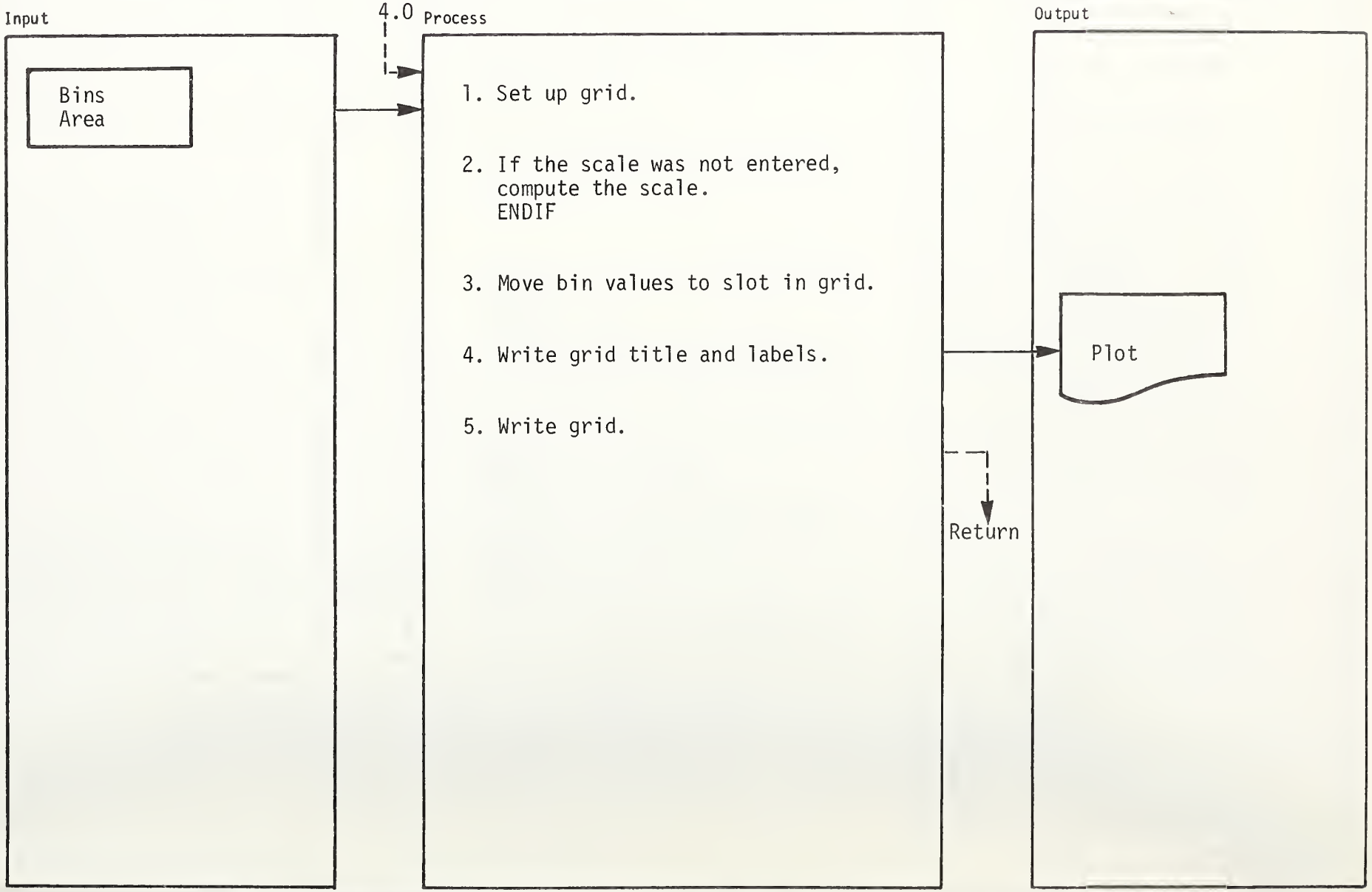
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 4.4 Name: ZLIST (Section 2.3.2.6) Description: Produce Statistical Summary and List



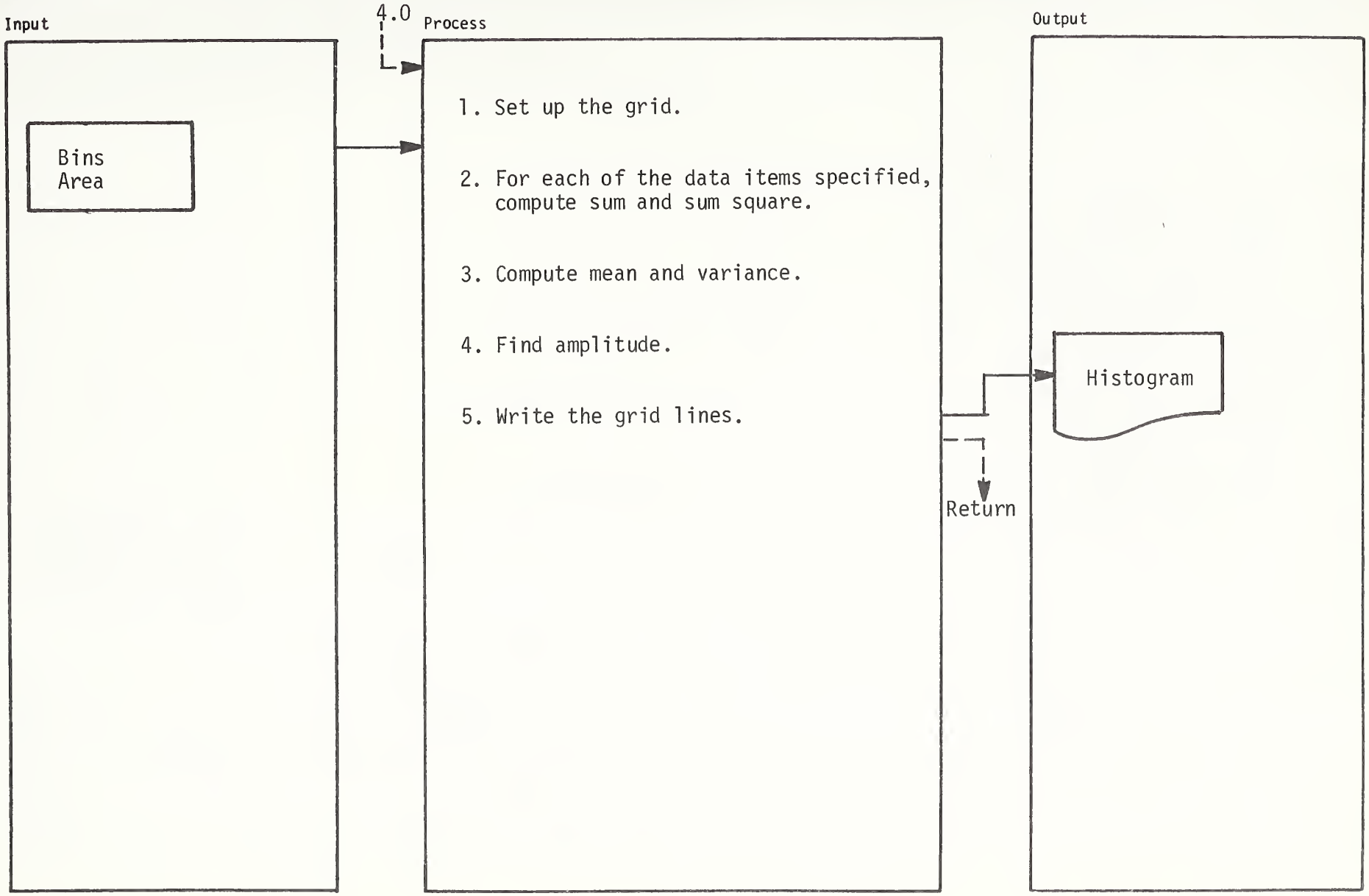
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 4.5 Name: ZPLOT (Section 2.3.2.6) Description: Produce a Time-Series Printer-Plot



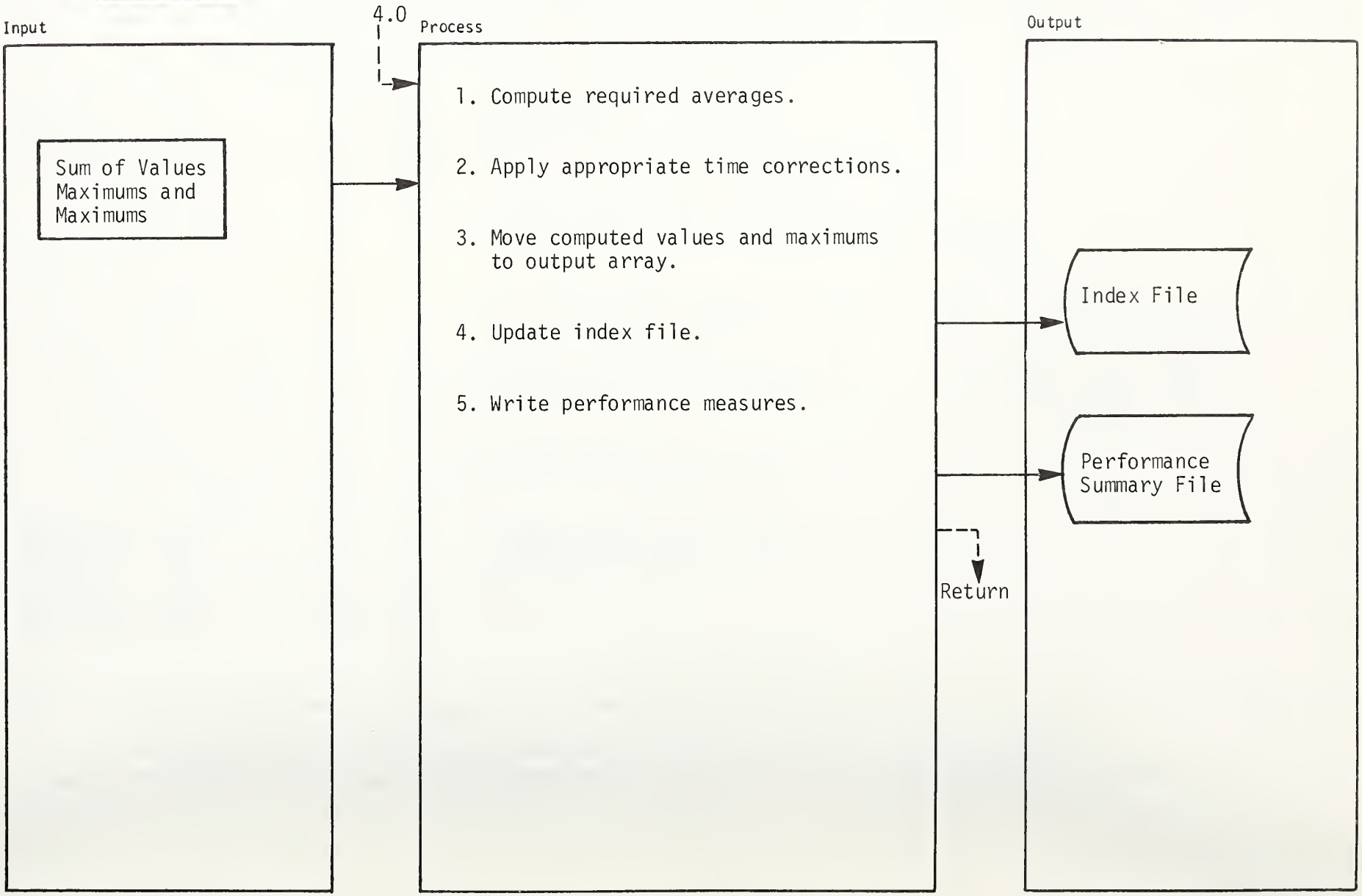
Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID: 4.6 Name: SZHIST (Section 2.3.2.6) Description: Produce a Histogram



Author: _____ System/Program: AGT-SOS/DSM Date: _____ Page: 1 of 1

Diagram ID 4.7 Name: SOPSUM Description: Performance Summary



APPENDIX C

REPORT OF NEW TECHNOLOGY

The Detailed Station Model (DSM) provides operational and performance measures of alternative station configurations and management policies with respect to vehicle and passenger handling capabilities. This model is integrated into the set of System Operations Studies models developed under this contract DOT-TSC-1220; it accepts as input a stream of vehicle arrivals at a specified station computed by the Discrete Event Simulation Model. Individual passenger detailed flow is eschewed in favor of passenger transit times at queues.

☆U.S. GOVERNMENT PRINTING OFFICE : 1982—500-797/325

100 copies

partment
sportation

arch and
ial Programs
inistration

Square
ge Massachusetts 02142

Official Business
Penalty for Private Use $300

Postage and Fees Paid
Research and Special
Programs Administration
DOT 513

U.S. MAIL