THE EVOLUTION OF A MODEL: Computational Aid, Policy Making, Planning and Scheduling

William S. Bonnell
Management Services Division
Eastman Kodak Company
Rochester, New York

ABSTRACT

This paper describes the development of a limited resource, backward scheduling, network model for an assembly department using DYNAMO. The model evolved in three stages: a calculation device, a policy exploration tool and a planning and scheduling system. An interesting feature of the model is the representation of the complex flow through various disassembly operations. Graphics and report interfaces with DYNAMO are discussed. The enclosed programs are provided on an as-is basis, without warranty either express or implied. No assurance of successful installation can be given.

INTRODUCTION

Originally this model of an assembly operation used DYNAMO solely as a calculation tool to provide rapid answers to changes in assumptions about reject rates, desired schedule and structure. It was simply a set of algebraic equations relating the variables of interest. There were no levels or rates to cause dynamic behavior. The dynamics were the result of exogenous variables.

A better representation was needed. Lead times, pipelines, decreasing reject rates and improvements in productivity during the start up phase of this product would cause interesting dynamic phenomena. Inventory policies and delays were introduced into the model. Disassembly and rework logic in various stages of manufacture was built into the model. Resource availabilities were incorporated as constraints on the various operations. Resource capacity was influenced by overtime policy and the number of work days per week. The model was constructed to allow these interactions to be explored and the consequences of policy decisions to be measured. A quantity called delivery backlog, the difference between cumulative schedule and cumulative deliveries, measured the performance of the system.

Having finished the start up phase, the model is implemented as a scheduling system. Each schedule period, the inventories and schedule are updated, and the model is run to determine the quantities each operation should produce each period for the next year. The upstream suppliers are modeled as potential constraints. The model causes each operation to produce the amount which, subject to its capacity constraint, meets the schedule, makes up for losses and fills each inventory to its desired level. When a constraint is reached on an operation, feedback through upstream and downstream inventories causes related operations to reduce their output.

As a sidelight, PL/I has been used to create a graphic interface and a custom report generator using the data saved in the DYNSAVE file. Graphic plots and custom reports make the communication of model results to management much easier.

The product structure and operations for the assembly department are shown in Figure 1. The finished assembly is made from a subassembly and a sleeve. The subassembly consists of three manufactured parts and a bearing. Parts 1, 3 and 4 have similar routings - punch, bend and paint. Part 2 is punched and painted. All operations are performed on automated equipment. Each operation has a QC check. Every operation can be characterized by input, output, disassembly, repair, scrap and return to stock flows. Some of these flows may be zero for the particular operation.

THE FIRST MODEL

The first model used DYNAMO auxiliary equations to represent the algebraic relationships of the department flows. The macro described in Figure 2 is for any department operation. Modeling the department consisted of choosing the right parameters for a particular operation and linking the input, output and return flows to other operation macros. This model assumed that all rates are some multiplier times current demand. There were no lead times or delays in the model.

It required about two weeks to build and calibrate the model. The department experimented with changing schedules and parameters to determine various rates. These rates were used to make estimates of personnel and machine resources. DYNAMO produced printed and plotted results. RERUN mode provided quick turnaround of parameter and schedule changes. The model met its goal.

THE SECOND MODEL

The first model generated questions which required more structure to answer.

Can the proposed transfer schedule be satisfied by proposed capacity availability? When and how much overtime will be required during the ramp up phase of the product? What are the effects of building inventory and filling pipelines during the ramp up? What if machines are late or don't produce as well as expected? How far behind will transfers lag and how long will it take to catch up? How much will accumulate in the disassembly operations over time for various levels of staffing?

Figure 3 is the flow chart for the second and third model. Levels are placed between operations where production decisions are required. Delays are used to represent operations such as disassembly and repair. The flow of disassembly is complex. Rejected final assemblies may be repaired. Those not repaired are taken apart into subassemblies. If subassemblies are good, they return to inventory. If they are bad, they may be scrapped or disassembled into components. The components may be returned to inventory, scrapped or returned to parts. Parts are reworked and returned to the appropriate inventory. Bending may create waste and rework. The steady state solution for this flow is tedious. It is important, however, to insure the integrity of the model. The procedure for obtaining the steady state solution is outlined in Appendix A. This solution also produces the set of scale factors used in formulating the first component of the desired rate equations for each operation.

The rate equations for most operations in the model are similar in structure. The transfer schedule is adjusted for the lead time of the operation. This schedule is scaled up for losses downstream. An inventory correction term, proportional to the desired value less actual value of the immediate downstream inventory, is added. The total rate is subject to availability of upstream components and resource capacity of the operation. Refer to Listing 1 (p. 29), an extract of model equations, for understanding of assembly rate (LAR). Figure 4 is a detailed representation of the variables affecting LAR. MIM5 is the positive minimum of 5 arguments. MCAP is a macro for capacity of an operation based on design speed, number of shifts and productivity index. Desired inventory, DLAI, is the desired buffer time multiplied by the current inventory outflow rate.

Leading demand, DMD(I), is calculated by using TIME + LEAD(I) as the entry argument for the DEMAND table function. LEAD(I) is the lead time for operation I. It is approximated by the sum of desired inventory times and pipeline delay times between the operation and the transfer rate. It is actually longer than this because of recycling. This small error is compensated by the inventory error correction term of each rate equation. The table is sampled every four weeks to match the transfer schedule.

Delivery backlog is a measure of schedule performance. Actual deliveries reduce the backlog. The transfer schedule increases it. Backlog, DLBKLG, is the cumulative schedule discrepency.

Capacities in the system may cause an operation to produce less than its desired rate. This will reduce downstream inventory. If severe enough, the inventory shortage will cascade through the system and cause deliveries not to meet transfer schedule. The resulting backlog puts pressure on the desired test rate, DFTR. As soon as capacity or inventory is available, the final test rate produces enough to drive the backlog to zero. Other rates in the system are trying to reduce their inventory error component to zero. Thus, if capacities are sufficient, each operation will produce enough to meet schedule, make up for downstream losses and fill pipelines and inventories to the desired level.

When a capacity constraint is limiting, the upstream inventory grows. This growth causes the inventory error term to reduce the next upstream rate. The system balances itself to

the limiting operation. Because all the rates are being driven from the same master transfer schedule, the system is stable.

Before the model, capacity requirements were based on estimated raw material usage. The recycling of parts was ignored. In the early stages of start up, when capacity was likely to be a problem, reject rates and recycling were high. This was not the time to ignore these factors. The model was able to explore the effect and give a much better estimate of required capacity.

RESULTS

Figures 5 through 11 describe the results of an experiment using typical but fabricated data. Figure 5 shows the utilization of the two available painting machines over the 60 weeks of this run. SHIFTS is a variable between 15 and 21 shifts per week. It stays at 15 until two painting machines are not able to meet the desired workload. At that point, it is free to increase up to 21 shifts. At 21 shifts, if capacity is not sufficient, painting would become a bottleneck operation. Other operations in the system may already be bottlenecked. Painting load would be paced by these operations. In this run, bending of part 3 is the bottleneck, as will be shown.

Figure 6 shows the shifts per week worked by the bending and assembly operations. These shifts are based on their respective actual rate equations. Bending of part 3 is limiting and begins working 21 shifts per week around week 12. As bending increases, the demand on other operations increases until week 55, when overtime is reduced on bending of part 3.

Figure 7 shows the various bending operation rates. The saw-tooth behavior is due to step increases in schedule instantaneously raising the desired inventory. Improvements in machine efficiency and reject factors also have some effect. Part 3 does not show the saw-tooth behavior, because it is being constrained by PMCAP, part 3 bending capacity.

Figure 8 shows insufficient part 3 bending capacity. Note how desired bending is above actual. Actual is being held to a capacity constraint. Its downstream inventory is less than desired. This raises the desired inflow rate. The inventory shortage cascades downstream until it affects transfers about week 26. The backlog increases, reflecting the discrepency between actual and desired deliveries. When part 3 bending capacity improves, the backlog is reduced. This occurs about week 42. Part 3 finished inventory, PFI, does not approach desired levels until the backlog has been eliminated and the

downstream pipelines filled. About week 52, enough capacity exists to start improving the part 3 inventory situation.

Figure 9 shows how the part 3 bending constraint passes through the system to affect other operations. Figure 9 graphs the assembly rates and their constraining factors. Final test and assembly rates (FTR and LAR) are being constrained by their upstream inventories from week 25 to week 50. FCAR, subassembly rate, is constrained by its capacity equation, FCCAP, from week 40 to week 50. At the same time that backlog disappears, inventory starts to build and the inventory constraints are lifted.

Figure 10 shows graphically the concept of delivery backlog. Actual transfers, LDR, and desired transfers, DMAND, are mismatched. When DMAND is greater than LDR, backlog grows. When LDR is greater, backlog decreases. Backlog is the accumulation of the mismatch.

Figure 11 is the total disassembly operation work load. Dividing this figure by the number of persons performing disassembly tells how many shifts per week are required. Disassembly peaks and decreases due to transfer requirements leveling and improvements in efficiencies and quality.

These are representative of the types of questions which the model has answered for the department. Because the model represents the department in great detail, it is a good device for exploring many types of "what if" questions.

THE THIRD MODEL

The third model has refinements to make user interaction easier. Table functions have offsets in the X argument based on the start week entered in the user data. This shifts the tables in time so they do not have to be updated unless the actual assumptions change. The data required from the user for scheduling is grouped at the end of the model. This will be put in a RERUN data file to be used with a compiled model in the near future. A report interface produces a document that looks like a schedule. It has descriptions rather than variable acronyms down the side. Time is across the top of the report. See Figure 12 for an example of the schedule produced.

Each period (4 weeks) the current inventories are put into the model and the schedule is revised to reflect the next 13 periods of transfers. The start date is updated, and the model is run. Period requirements for the next year for each operation, as well as inventory projections, are generated. The

period requirements are integrated rates which are zeroed at the beginning of each period by the CLEAR function (Listing 1). An example of its use is the variable SCHED.

The DYNSAVE file produced by the SAVE statement is a good means of interfacing DYNAMO to plotting and report writing programs. Listing 2 is an example of a PL/I program which reads the DYNSAVE file and a report format file to produce custom reports. Listing 3 is the sample format file which produces the schedule for the department. Descriptions in the format file replace variable acronyms in the output when the associated variable is found in the DYNSAVE file. The data is formatted across the page. The decimal point of the data is controlled by the scale factors (zero through five or blank) in column 1Ø. Line spacing and centered titles are also available.

A graphic interface is shown in Listing 4. The principle is the same as the report writer. The DYNSAVE file is produced by the model with the appropriate variables using SAVE and SAVPER. This file and a plot file are read by the program. The plot file contains graphics statements required by the particular graphics package. It also contains one or more statements of variables to be plotted as follows:

```
USE TIME=X SCHED=Y .
USE TIME=X ACTUAL=Y .
```

TIME, SCHED and ACTUAL are looked up in the DYNSAVE file and substituted in the output file along with the appropriate graphics statements. The output file is directly readable by the graphics package and produces the plot of SCHED and ACTUAL versus TIME.

Other interfaces have been built for statistical analysis packages using the same principle.

SUMMARY

This model has been successful. It is still in use making calculations, formulating policy and scheduling. The department industrial engineer has been trained in DYNAMO and is able to make modifications. The scheduler has sufficient understanding to produce the schedule.

APPENDIX A

STEADY STATE SOLUTION AND RATE EQUATIONS

To get the steady state solution for levels in the system, set inflow rates equal to outflow rates. Solve for rates in terms of demand rate, DMD(1) and system constants. These rates times the upstream desired inventory buffer times are the initial steady state level values. When these levels are used in the model and all exogenous inputs are constant, none of the model variables should change. If this is not the case, within the accuracy of single-precision variables, there is an error in formulation. The rigors of getting a steady state solution should be performed whenever the structure of the model has been changed. This helps insure the integrity of the model.

The solution for the rates in terms of demand provides the first component of the dynamic rate equations. Substitute leading demand, DMD(I), for DMD(1) and add the downstream inventory correction term.

An example of this process follows:

EXAMPLE

(REFER TO FIGURE 3)

FTR*(1-FTRF)=LDR = DMD(2) NOTE the suffix F means fraction and corresponds to the rate with suffix R on Figure 3.

1. FTR=DMD(2)/F1 where F1=1-FTRF

==================================

LAR*(1-LRF)+LAR*LRF*LRGF=FTR-FTRGR

2. FTRGR=FTR*FTRF*FTRGF

substitute 1. and 2. then solve for

3. LAR=DMD(3)*R1Ø/(F1*F2)

where F2=1-LRF+LRF*LRGF AND

R1Ø=1-FTRF*(1-FTTF)*FTRGF

==================================

FCAR*(1-FCRF)+LTFCRR=LAR

4. LTFCRR=(LAR*LRF*(1-LRGF)+(FTR*FTRF*FTTF+FTR*FTRF*(1-FTTF)*(1-FTRGF))*(1-FSRF))*LTFCRF

substitute 1.,2.,3. and 4. then solve for

5. FCAR=DMD(4)*R1Ø*R2/(F1*F2*F3)

where R2=(1-LTFCRF*(LRF*(1-LRGF)+F2*R1/R1Ø))

and F3=(1-FCRF)

and R1=FTRF*(FTTF+(1-FTTF)*(1-FTRGF))*(1-FSRF)

===============================

MLSAR*(1-MARSF)+FCMCR=FCAR

6. FCMCR=(FCAR*FCRF+LTFCTR)*(1-FCMTSF)*(1-FCMMSF)

7. LTFCTR=(FTR*R1+LAR*LRF*(1-LRGF))*LTFCTF

substitute 1.,3.,5.,6.,7. and solve for

MLSAR=DMD(8)*R1Ø*R3/(F1*F2*F3*F4)

where R3=R2*(1-F18*(FCRF+LTFCTF/R2*(R1*F2*F3/R1Ø+F3*LRF*(1-LRGF))))

and F18=(1-FCMTSF)*(1-FCMMSF)

and F4=(1-MARSF)

Figure 1
Product Structure
and
Operations

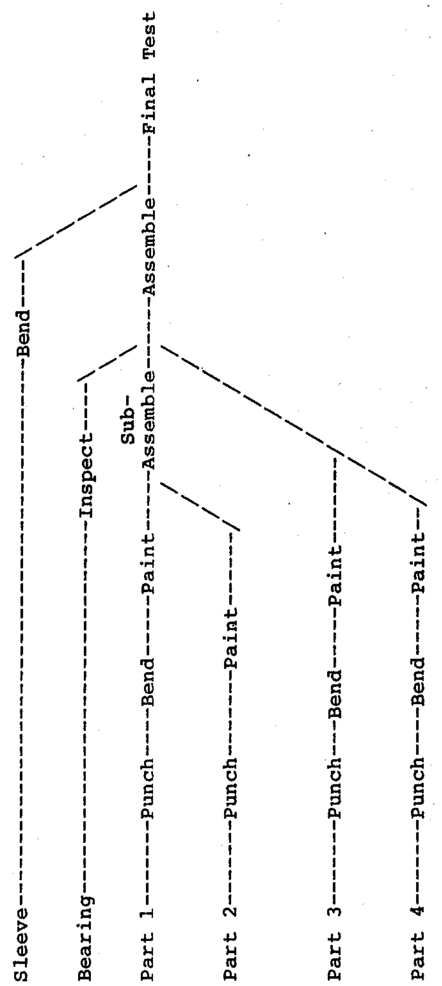
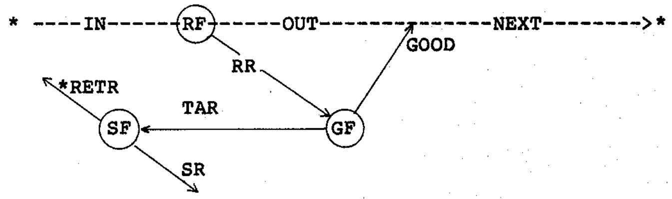


Figure 2

OPERATION MACRO

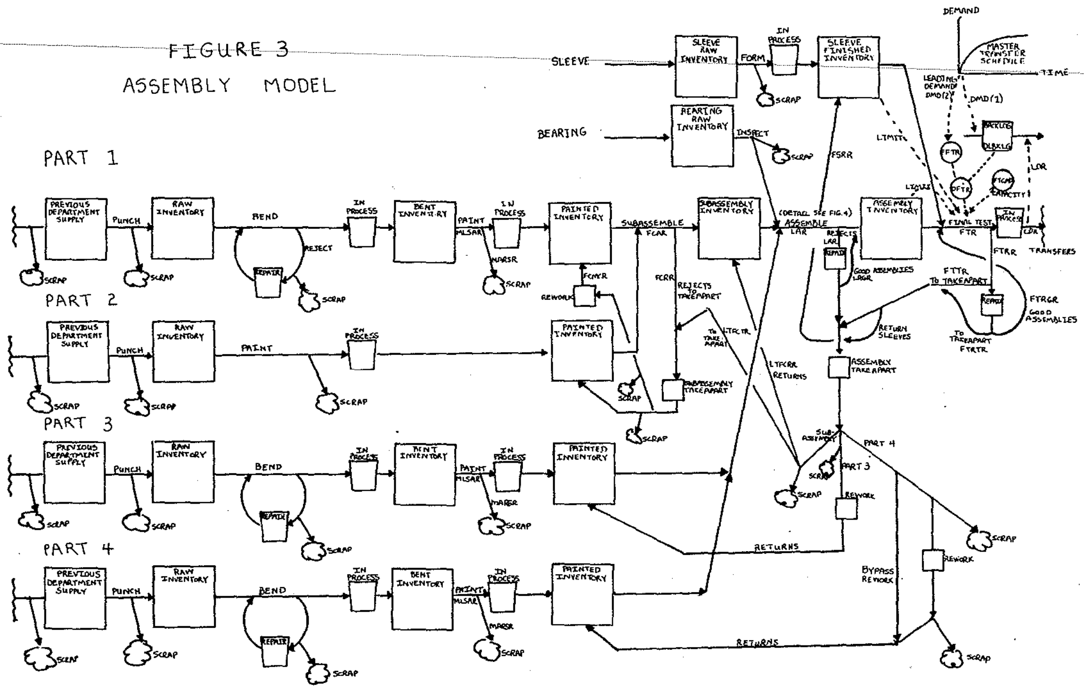


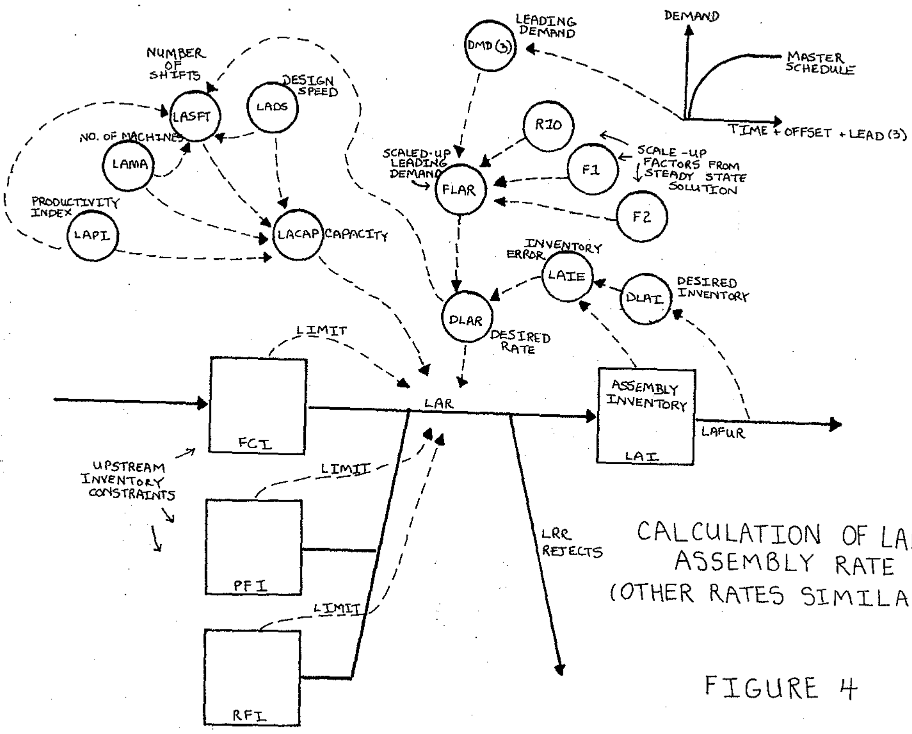
* connect to other MACROS

```
MACRO   IN(RF,NEXT,RR,GF,GOOD,OUT,TAR,SF,SR,RETR)

A IN=NEXT/(1-RF+RF*GF)    input from previous MACRO
A OUT=IN*(1-RF)           output to next MACRO
A RR=IN*RF                reject rate
A GOOD=RR*GF              good rate
A TAR=RR*(1-GF)           takeapart rate
A SR=TAR*SF               scrap rate
A RETR=TAR*(1-SF)         return rate

NOTE    RF is the reject fraction

NOTE    SF is the scrap fraction

NOTE    GF is the good fraction
```

FIGURE 3

ASSEMBLY MODEL





CALCULATION OF LAR ASSEMBLY RATE (OTHER RATES SIMILAR)

FIGURE 4

FIGURE 5
PAINTING MACHINES AND SHIFTS REQUIRED

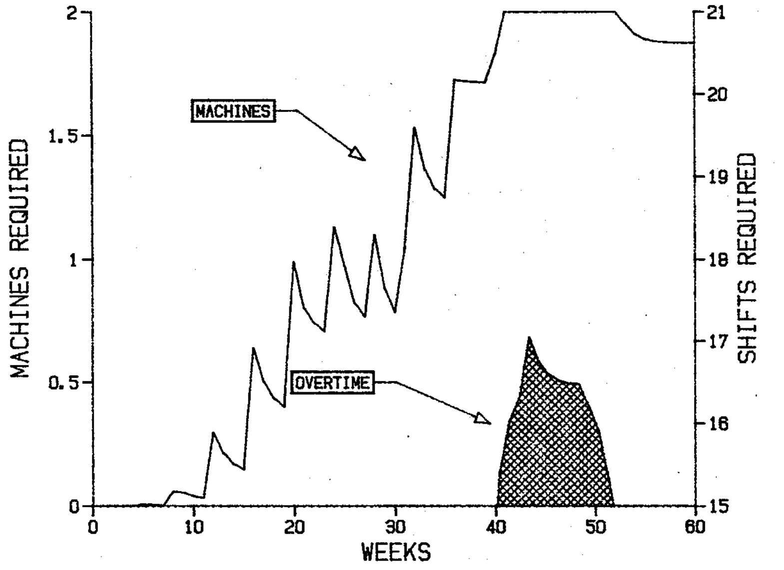


FIGURE 6
ACTUAL SHIFTS

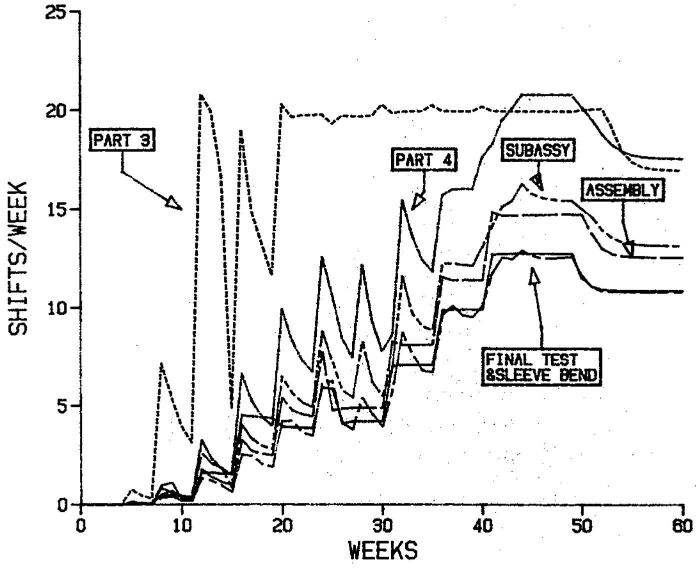


FIGURE 7
BENDING RATES

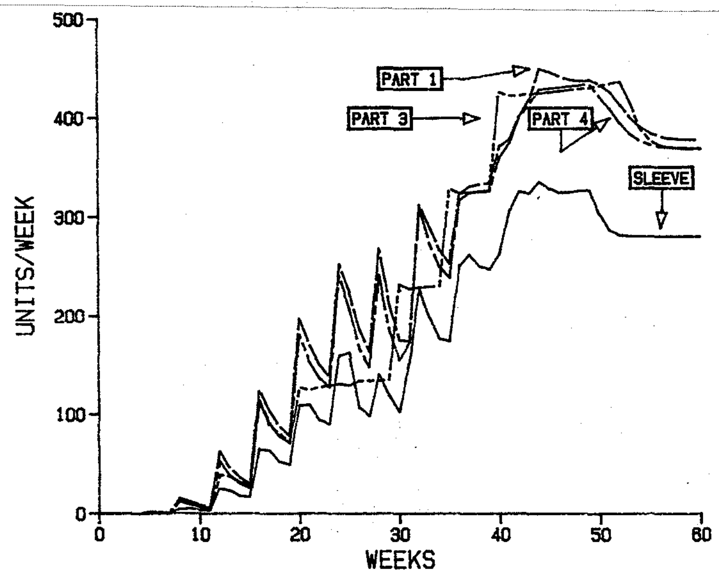


FIGURE 8
PART 3 BENDING
INSUFFICIENT CAPACITY

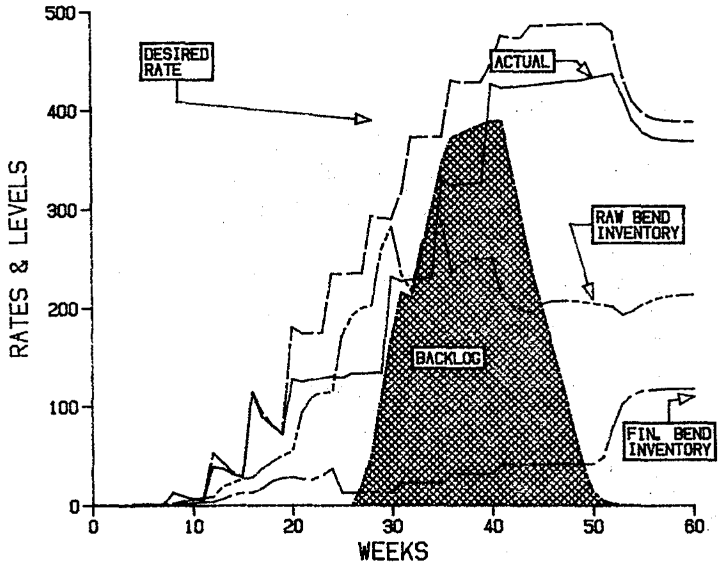


FIGURE 9
CONSTRAINED RATES
BY CAPACITY OR INVENTORY

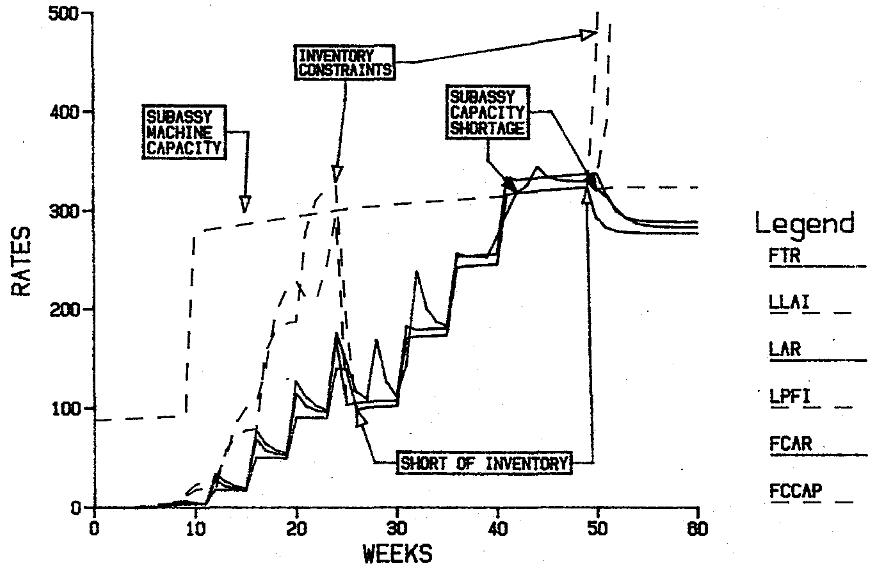


FIGURE 10
TRANSFERS AND BACKLOG

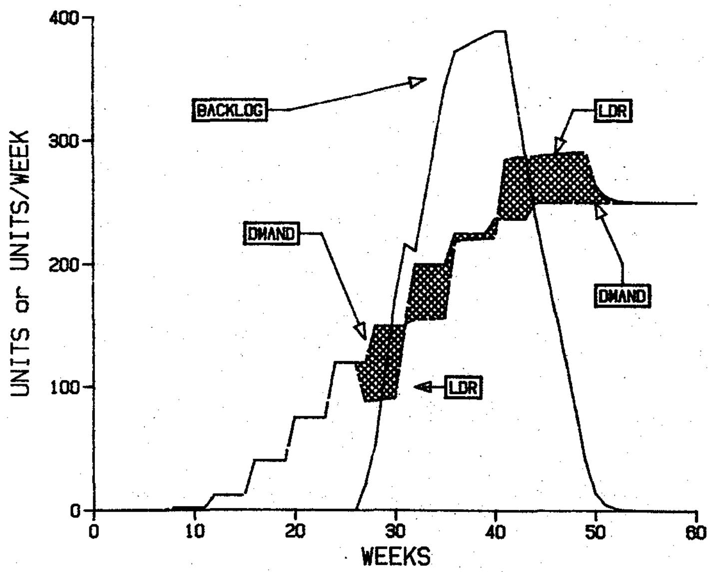


FIGURE 11
TOTAL DISASSEMBLY WORKLOAD

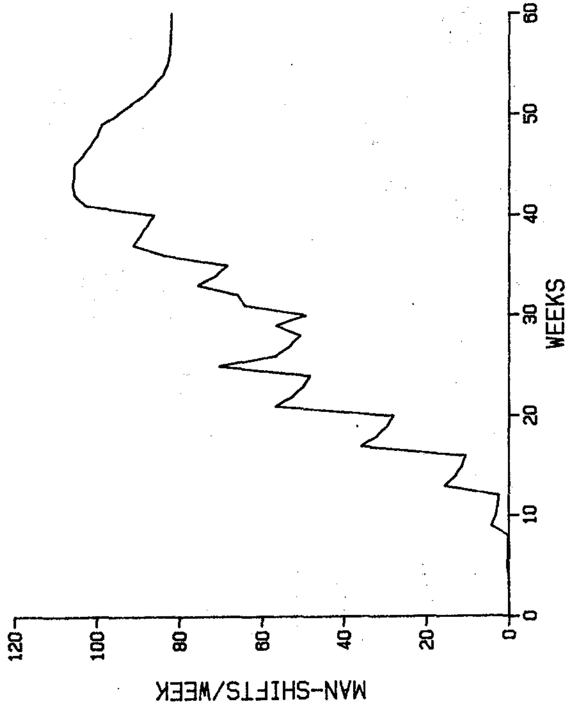


FIGURE 12

SAMPLE SCHEDULE PRODUCED BY REPORT WRITER

Year	1982	1982	1982	1982	1982	1982	1982
End of Period	0	1	2	3	4	5	6
Master Schedule	0	0	0	8	48	160	300
			Final Test				
Actual Deliveries	0	0	0	8	48	160	300
Backlog	0	0	0	0	0	0	0
Test WIP	0	0	1	4	13	24	42
Reject Rate	0	0	0	2	9	24	43
Final Test Rate	0	0	1	13	66	195	361
Final Takeapart	0	0	0	2	9	23	40
			Assembly				
Inventory	0	0	0	3	10	23	41
Good Assy Rate	0	0	1	16	73	208	376
Assembly Rejects	0	0	0	1	6	18	32
Assembly Rate	0	0	1	17	78	222	402
Assy Takeapart	0	0	0	1	5	14	25
Test & Assy T/A	0	0	0	3	13	36	65
Assy Scrap Rate	0	0	0	1	4	11	19
Pt 3 Scrap Rate	0	0	0	2	7	18	32
Pt 4 Scrap Rate	0	0	0	2	7	18	32
			Subassembly				
Subassy Inv.	0	0	1	4	13	28	47
Good Subassy Rate	0	0	2	18	81	219	388
Subassy Rejects	0	0	0	2	8	22	36
Subassy Rate	0	0	2	20	89	241	424
Subassy T/A	0	0	0	3	11	29	49
Subassy Scrap	0	0	0	1	5	14	24
Pt 2 T/A Scrap	0	0	0	1	5	14	24
			Part 1				
Pt 1 Painted Inv.	0	0	1	5	16	32	52
Good Pt 1 Paint	0	0	2	23	97	247	428
Pt 1 Paint WIP	0	0	0	1	2	7	12
Pt 1 Paint Scrp	0	0	0	4	15	38	65
Pt 1 Paint Rate	0	0	3	27	114	289	498
Pt 1 Bend Inv.	0	0	1	6	21	38	61
Good Pt 1 Rate	0	0	4	33	128	306	521

```
NOTE                          LISTING 1

                   EXTRACTED FILE OF DYNAMO STATEMENTS
                     (GOES WITH FIGURE 4 AND PAGE 6)
MACRO MIM5(A,B,C,D,E)    MIN OF 5 ARGUMENTS
     A MIM5.K=MAX(0,MIN(A.K,MIN(B.K,MIN(C.K,MIN(D.K,E.K)))))
MEND
MACRO SHIFTS(DESRAT,PINDEX,DS,MACH)  SHIFTS NEEDED FOR DESIRED RATE
     A $DESFT.K=DESRAT.K/MAX(1E-10,PINDEX.K*DS.K*MACH.K)
     A SHIFTS.K=MAX(15,MIN(21,$DESFT.K))
MEND
MACRO MCAP(CAPSW,PINDEX,DS,SHIFTS,MACH) MACHINE CAPACITY
     A MCAP.K=1E6*(1-CAPSW)+CAPSW*DS.K*PINDEX.K*SHIFTS.K*MACH.K
MEND
NOTE **** SHOW THE CALCULATION OF LAR -  ASSEMBLY RATE *****
A LAR.K=MIM5(DLAR.K,LACAP.K,RFI.K/DT,PFI.K/DT,FCI.K/DT)  ASSY RATE
NOTE    RFI,PFI,RCI ARE UPSTREAM LEVELS WHICH MAY BE CONSTRAINING
A   DLAR.K=MAX(FLAR.K+LAIE.K/(CORRT*F2.K),0) DESIRED  ASSY RATE
A     FLAR.K=R10.K*DMD.K(3)/(F1.K*F2.K)    SCALED-UP LEADING ASSY RATE
A       F1.K=1-FTRF.K     SCALE UP FACTOR
A       F2.K=(1-LRF.K*(1-LRGF))     SCALE UP FACTOR (LAR)
A       R10.K=(1-FTRF.K*FTRGF.K*(1-FTTF.K)) RECYCLE FACTOR FINAL TEST
A     LAIE.K=DLAI.K-LAI.K      ASSY INVENTORY ERROR
A     LACAP.K=MCAP(CAPSW,LAPI.K,LADS,LASFT.K,LAMA.K)  ASSY CAPACTIY
C       LADS=7.58     ASSY. DESIGN SPEED
A       LAMA.K=1+STEP(1,8-OFFSET)+STEP(2,15-OFFSET)  ASSY MACHINES
A       LASFT.K=SHIFTS(DLAR.K,LAPI.K,LADS,LAMA.K) SHIFTS TO DO DLAR
A       LAPI.K=TABHL(ILAP,TIME.K+OFFSET,0,52,26)PRODUCTIVITY INDEX
T        ILAP=.65,.72,.76 TABLE OF  ASSY PRODUCTION INDEX

R  LRR.KL=LAR.K*LRF.K        REJECT RATE
A   LRF.K=TABHL(LRFT,TIME.K+OFFSET,0,36,4)    REJECT FRACTION
T    LRFT=.110,.090,.08,.08,.08,.08,.08,.08,.06,.06 REJECT TABLE

L LAI.K=LAI.J+DT*(LGR.JK-LAFUR.J)        ASSY INVENTORY
N   LAI=0   INITIALLY EMPTY
A  DLAI.K=LAFUR.K*LAIT.K     DESIRED  ASSY INVENTORY
A   LAIT.K=INVT.K     DESIRED  ASSY INV BUFFER TIME
A  LAFUR.K=FTR.K-FTRGR.K      ASSY FINISHED USAGE RATE
R  LGR.KL=LAACR.K+LRGR.K       GOOD RATE
A   LAACR.K=LAR.K*(1-LRF.K)     ASSY ACCEPT RATE
A   LROR.K=DEL1P(LRR.JK,LARDT,LARD.K,-1)     REPAIR OUTPUT RATE
A   LRGR.K=LROR.K*LRGF     REPAIR GOOD RATE
C     LRGF=.20      REPAIR GOOD FRACTION

A DMD.K(I)=SAMPLE(TABHL(DEMAND,TIME.K+LEAD(I),0,52,4),4,DEMAND(1))
    DEMAND AT LEAD TIME 'I' WHERE 'I' REPRESENTS AN OPERATION
L DLBKLG.K=DLBKLG.J+DT*(DMD.J(1)-LDR.J)       DELIVERY BACKLOG
N   DLBKLG=0
A   DFTR.K=MAX(0,FFTR.K+(DLBKLG.K/(CORRT*F1.K))) DESIRED FINAL TEST RATE
L SCHED.K=SCHED.J+DT*(DMD.J(1)-CLEAR.J*SCHED.J)      INTEGRATED DEMAND
   AVERAGE VALUE OF SCHED OVER 4 WEEK PERIOD
N  SCHED=0
A   CLEAR.K=PULSE(1/DT,.01,4)    RESETS INTEGRATED ACCUMULATIONS EACH PD
```



```
/*                          LISTING 2

        THIS PROGRAM TAKES THE OUTPUT OF THE DYNAMO MODEL STORED IN
        THE DYNSAVE FILE AND REFORMATS IT TO HAVE TIME ACROSS THE TOP
        AND VARIABLES DOWN THE SIDE.  THE VARIABLES, TITLES AND
        FORMATTING INFORMATION ARE STORED IN A FILE CALLED MODEL DESC .
        THE VALUES OF THE VARIABLES ARE STORED IN A FILE CALLED
        RUN DYNSAVE.  IN THE MODEL FILE CALLED MODEL DYNAMO THE
        VARIABLES TO BE SAVED ARE PUT ON A SAVE, CPRINT OR CPLOT CARD
        AND SAVPER MUST HAVE A POSITIVE VALUE MULTIPLE OF DT.  THE
        OUTPUT IS STORED IN FILENAME REPORT.

                          INPUT FILE FORMAT
 COL 1 - VARIABLE NAME   COL 10 - SCALING NUMBER  COL 15 - DESCRIPTION
    6 CHARACTERS               BLANK OR 1-5             65 CHARACTERS
*/
REFORM: PROC OPTIONS (MAIN);
        ON ENDFILE(INPUT) EOFI='1'B;
        ON ENDFILE(DESC) EOFD='1'B;
        DCL EOFD BIT(1);
        DCL EOFI BIT(1);
        DCL VARNAME(200) CHAR(7);
        DCL PLACES(200) FIXED BIN;
        DCL DESCRPT(200) CHAR(65) ;
        DCL NAM(NVAR) CHAR(4) CTL;
        DCL E(NVAR) CHAR(4) CTL;
        DCL NAME(NVAR) CHAR(7) CTL;
        DCL NVAR FIXED BIN(31);
        DCL IN CHAR(1060) ;
        DCL IN1 CHAR(1060) ;
        DCL IN2 CHAR(1060) VARYING;
        DCL ARRAY(200,NVAR) FLOAT CTL;
        DCL VECTOR(NVAR) FLOAT CTL;

        I=0; /* READ FORMAT FILE */

        DO WHILE (^EOFD);
          I=I+1;
         GET FILE(DESC) EDIT(VARNAME(I),PLACES(I),DESCRPT(I))
             (COL(1),A(7),COL(10),F(1),COL(15),A(65));
        END;
        ENTRIES=I-1;

        READ FILE(INPUT)IGNORE(1); /* READ DYNSAVE FILE
        READ FILE(INPUT)INTO(IN2);
        NVAR=LENGTH(IN2)/4;
        ALLOCATE NAM,E,NAME,ARRAY,VECTOR;
        IN=IN2;
        READ FILE(INPUT)INTO(IN2);
           IN1=IN2;
           DO J=1 TO NVAR;
           NAM(J)=SUBSTR(IN,4*J-3,4);
           E(J)=SUBSTR(IN1,4*J-3,4);
           NAME(J)=SUBSTR(NAM(J),2,3)||E(J);
```

```
          END;
       I=0;
       DO WHILE (^EOFI);
          READ FILE(INPUT)INTO(VECTOR);
          I=I+1;
          DO J=1 TO NVAR;
             ARRAY(I,J)=VECTOR(J);
          END;
       END;
       LAST=I-3;

DO MATCH=1 TO ENTRIES; /* PRINT RESULTS */
   IF VARNAME(MATCH)='TITLE' THEN DO;
      X=66-INDEX(DESCRPT(MATCH),'   ')/2;
      PUT EDIT(DESCRPT(MATCH))(COL(X),A);
      GOTO NEXT;
   END;

   IF VARNAME(MATCH)='SKIP' THEN DO;
      IF PLACES(MATCH)=0 THEN PUT SKIP(2);
      ELSE PUT SKIP(PLACES(MATCH)+1);
      GOTO NEXT;
   END;

   DO J=1 TO NVAR; /* TABLE LOOKUP */
   IF VARNAME(MATCH)=NAME(J) THEN GOTO OUTPUT;
   END;

   GOTO NEXT; /* NOT FOUND */

OUTPUT:  PUT EDIT(DESCRPT(MATCH))(COL(1),A(17));
   IF PLACES(MATCH)=0 THEN
   PUT EDIT((ARRAY(I,J) DO I=1 TO LAST))(F(8));
   ELSE IF PLACES(MATCH)=1 THEN
   PUT EDIT((ARRAY(I,J) DO I=1 TO LAST))(F(8,1));
   ELSE IF PLACES(MATCH)=2 THEN
   PUT EDIT((ARRAY(I,J) DO I=1 TO LAST))(F(8,2));
   ELSE IF PLACES(MATCH)=3 THEN
   PUT EDIT((ARRAY(I,J) DO I=1 TO LAST))(F(8,3));
   ELSE IF PLACES(MATCH)=4 THEN
   PUT EDIT((ARRAY(I,J) DO I=1 TO LAST))(F(8,4));
   ELSE IF PLACES(MATCH)=5 THEN
   PUT EDIT((ARRAY(I,J) DO I=1 TO LAST))(F(8,5));
   ELSE PUT EDIT((ARRAY(I,J) DO I=1 TO LAST))(E(8,1));
NEXT:  END;
DONE: END REFORM;
```


LISTING 3

```
TITLE        FIGURE 12
SKIP
TITLE        SAMPLE SCHEDULE PRODUCED BY REPORT WRITER
SKIP
YEAR         Year
PERIOD       End of Period
SCHED        Master Schedule
SKIP
TITLE        Final Test
ILDR         Actual Deliveries
DLBKLG       Backlog
FTRIP        Test WIP
IFTRR        Reject Rate
IFTR         Final Test Rate
IFLCR        Final Takeapart
TITLE        Assembly
LAI          Inventory
ILGR         Good Assy Rate
ILRR         Assembly Rejects
ILAR         Assembly Rate
ILRCR        Assy Takeapart
ILTOR        Test & Assy T/A
ILFSR        Assy Scrap Rate
ILTPSR       Pt 3 Scrap Rate
ILTRSR       Pt 4 Scrap Rate
TITLE        Subassembly
FCI          Subassy Inv.
IFCGR        Good Subassy Rate
IFCRR        Subassy Rejects
IFCAR        Subassy Rate
IFCTAR       Subassy T/A
IFCMSR       Subassy Scrap
IFCASR       Pt 2 T/A Scrap
TITLE        Part 1
MARFI        Pt 1 Painted Inv.
IMARCR       Good Pt 1 Paint
MARIP        Pt 1 Paint WIP
IMARSR       Pt 1 Paint Scrp.
IMLSAR       Pt 1 Paint Rate
MMFI         Pt 1 Bend Inv.
IMMCR        Good Pt 1 Rate
```

```
/*                          LISTING 4


      THIS PROGRAM READS RUN DYNSAVE AND FILE DYNPLOT AND CREATES
   FILE DATA. FILE DATA CAN BE READ DIRECTLY BY THE GRAPHICS PACKAGE AND
   PLOTTED ACCORDING TO THE INSTRUCTIONS IN THE FILE DYNPLOT.
*/


DYNTAG: PROC OPTIONS(MAIN);
        DCL (X$,Y$) CHAR(72) VARYING;
        DCL A$ CHAR(72) VARYING;
        DCL (EOF1,EOF2) BIT(1);
        DCL (IN80,OUT) CHAR(80);
        DCL NUM CHAR(8) DEF OUT POS(73);
        DCL ID CHAR(3) DEF(IN80);
        DCL (XVARID,YVARID) CHAR(7) ;
        DCL (START,END) FIXED BIN(31);
        DCL (XVAR#,YVAR#) FIXED BIN(31);
        DCL NAM(100) CHAR(4) ;
        DCL E(100) CHAR(4);
        DCL NAME(100) CHAR(7);
        DCL NVAR FIXED BIN(31);
        DCL Q FIXED BIN(31);
        DCL IN CHAR(1744) ;
        DCL IN1 CHAR(1744) ;
        DCL IN2 CHAR(1744) VARYING;
        DCL ARRAY(200,NVAR) FLOAT CTL;
        DCL VECTOR(NVAR) FLOAT CTL;
        DCL IOBS FIXED BIN(31);
        DCL (I,J,K,L) FIXED BIN(31);
        ON ENDFILE(INPUT)
        EOF1='1'B;
        ON ENDFILE(DYNPLOT)
        EOF2='1'B;
        READ FILE(INPUT) IGNORE(1);
        READ FILE(INPUT) INTO(IN2);
        NVAR=LENGTH(IN2)/4;
        ALLOCATE ARRAY,VECTOR;
        IN=IN2;
        READ FILE(INPUT) INTO(IN1); /* GET VARIABLE NAMES */
           DO J=1 TO NVAR;
           NAM(J)=SUBSTR(IN,4*J-3,4);
           E(J)=SUBSTR(IN1,4*J-3,4);
           NAME(J)=SUBSTR(NAM(J),2,3)||E(J);
           Q=INDEX(NAME(J),' ');
           NAME(J)=SUBSTR(NAME(J),1,Q-1);
           END;
        I=0;
           DO WHILE(^EOF1); /* FORM ARRAY OF DATA */
           READ FILE(INPUT) INTO(VECTOR);
              IF ^EOF1 THEN DO;
```



```
                 I=I+1;
                    DO J=1 TO NVAR;
                    ARRAY(I,J)=VECTOR(J);
                    END;
                 END;
              END;
           IOBS=I-2;
              DO WHILE(^EOF2); /* GET PLOTTING INSTRUCTIONS */
              READ FILE(DYNPLOT) INTO(IN80);
              IF ^EOF2 THEN DO;
                 IF INDEX(IN80,'USE')^=0 THEN
                    DO;
                    START=INDEX(IN80,'Y=')+2;
                    END=INDEX(SUBSTR(IN80,START),' ')-1;
                    YVARID=SUBSTR(IN80,START,END);
                    YVAR#=SEARCH(NAME,YVARID,NVAR);
                       IF YVAR#=0 THEN
                       PUT EDIT('DYNPLOT VARIABLE ',YVARID,' NOT IN DYNSAV
                       E')(COL(1),A,A,A);
                    START=INDEX(IN80,'X=')+2;
                    END=INDEX(SUBSTR(IN80,START),' ')-1;
                    XVARID=SUBSTR(IN80,START,END);
                    XVAR#=SEARCH(NAME,XVARID,NVAR);
                       IF XVAR#=0 THEN
                       PUT EDIT('DYNPLOT VARIABLE ',XVARID,
                       ' NOT IN DYNSAVE')(COL(1),A,A,A);
                       IF(YVAR#^=0&XVAR#^=0)THEN
                          DO;
                          OUT='"'||YVARID||'"';
                          NUM='        ';
                          WRITE FILE(TAGFILE) FROM(OUT);
                             DO I=1 TO IOBS;
                             X$=ARRAY(I,XVAR#);
                             Y$=ARRAY(I,YVAR#);
                             A$=X$||','||Y$||' ';
                             OUT=A$;
                             NUM='        ';
                             WRITE FILE(TAGFILE) FROM(OUT);
                             END;
                          END;
                    END;
                 ELSE DO;
                 OUT=IN80;
                 NUM='        ';
      IF INDEX(OUT,'.     ')=0 THEN
         SUBSTR(OUT,72,1)='.';
                 WRITE FILE(TAGFILE) FROM(OUT); /* CREATE TEL-A-GRAF */
                 END;
              END;
           END;
SEARCH:    PROC(NAME,VARID,NVAR); /* SEQUENTIAL SEARCH */
           DCL NAME(*) CHAR(*) ;
```

```
    DCL VARID CHAR(*) ;
    DCL (VAR#,NVAR,I) FIXED BIN(31);
    VAR#=0;
    I=1;
       DO WHILE(NAME(I)^=VARID&I<=NVAR);
       I=I+1;
       END;
       IF I<=NVAR THEN
       VAR#=I;
       ELSE
       VAR#=0;
    RETURN(VAR#);
    END SEARCH;
 END DYNTAG;
```