An analysis of stocking strategy in a multi-echelon assembly system by computer modelling

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:


Metadata Record: https://dspace.lboro.ac.uk/2134/32932

Publisher: © P.D. Laurence

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

Please cite the published version.
AUTHOR/FILING TITLE

LAURENCE, P. D.

ACCESSION/COPY NO.

007855/02

VOL. NO.

CLASS MARK

LOAN COPY
AN ANALYSIS OF STOCKING STRATEGY IN A MULTI-
ECHELON ASSEMBLY SYSTEM BY COMPUTER MODELLING

VOLUME II
APPENDIX B

FULL PROGRAM DESCRIPTION

The structure of the simulation model is described in detail to provide a more complete definition than in the main text and permit later maintenance and modification if required.

Reference should be made to

Appendix C - File Descriptions

and

Appendix D - Data Dictionary

for certain definitions.

The programs are defined by a narrative summary, logic flowchart, the program source listing and sample output prints where applicable.

CONTENTS

B 1 MASTER PROGRAM STOCKMODEL
B 2 ALLOCATE
B3 3 AQPRINT (LQPRINT; TQPRINT; SQPRINT)
B 4 ARRIVAL
B 5 DELPERF
B 6 DESPATCH
B 7 EBQ
B 8 EXPEDITE
B 9 EXPLODE
B10 FESALL
B11  FORECAST
B12  HSAMPLE
B13  INITODS
B14  ISSWIP
B15  LIMIT
B16  LOAD
B17  LOADPLAN
B18  MONTHEVENT
B19  NETTING
B20  NORMAL
B21  OFFLINE
B22  OHRESET
B23  OPFCREATE
B24  OPFPRT
B25  ORDBKPRT
B26  ORDERS
B27  PLNPRT
B28  PNMFCREATE
B29  PNMFPRINT
B30  QPLAN
B31  QPLANPRINT
B32  QTREVENTS
B33  QTRRESET
B34  RANDOM
B35  RECEIVE
B36  SCHED
B37  STKALL
B38  STOCKVAL
B39  SUBPROG
B40  TRANSLATE 3
B41  TRENDI
B42  WEEKEVENT
B43  WEEKLYPLAN
B44  WEEKPRINT
The master STOCKMODEL program performs five primary tasks:

a) definition of the probability density functions utilised by the model.
b) establishment of the parameters controlling each experiment, known as the run parameters.
c) selection of the seeds required by the pseudo random number generator.
d) establishment of the initial status of the model.
e) control of the simulation experiment.

1. Probability Density Functions

A total of twelve probability density functions are used by the model, being either frequency histograms defined by parameter and relative frequency, or normal distributions defined by the mean and co-efficient of variance. Histogram definitions are contained within the DATA statement, and normal distribution parameters are specified as an arithmetic expression.

No probability density function may change its characteristics during the execution of a simulation experiment, and the parameters may only be changed by altering the source program.

2. Run Parameters

The parameters controlling each experiment are presented to the program as external data at the start of each simulation experiment. The parameters may be classified as either simple or complex, according to their impact on the model logic.

Complex parameters I have defined as those substantially altering the processing logic. For example, the mode of manufacture may be specified as make to order only, make to stock only, or mixed production. The priority rules for selection of orders from each queue may be specified as first-in-first-out or due date priority.
Simple parameters are those which affect an element in an arithmetic expression but do not directly influence the processing logic. Examples include the buffer stock parameters, order book planning rules, schedule response period, capacity utilisation and order mix trends.

3. Random Number Seeds

The seeds required to initiate the pseudo random number streams are presented to the program as external data. The performance of each sampling activity is controlled by providing a separate seed for each activity and maintaining the identity of each seed throughout the simulation experiment.

The pseudo random number generator has been designed to accommodate negative seeds where antithetic sequences are required and, in some instances, will cause a switch from stochastic to deterministic mode on presentation of a zero seed.

4. Initial Conditions

Two types of initial conditions are required by the program. Certain parameters which control the internal operation of the model are internally initialised; examples include the timing mechanisms and performance indices. Externally specified initial conditions describe the status of the system files at the commencement of each experiment. Specification of the initial conditions fulfils two purposes. Firstly, to enable certain logical processes which are dependent upon previous periods, and secondly to reduce the time to achieve steady state conditions.

Wherever possible, the various system files are created from a common source of input data to ensure integrity of dispersed data. Initial conditions are classified as orders related, or parts related. Orders related files are all derived from the Orders Placed File (OPFILE) and include the queues (ALLQ, LINEQ, TESTQ), assembly and test work-in-progress (IFILE), order book (ORDBK) and total order load (TOTAL). The orders received history (ORDHIST) is created by simulating a twelve month order input pattern, using the order generation logic contained in subroutine ORDERS. This en-
sures consistency with the subsequent simulation experiment.

Parts related conditions include the line running rates (R) and the previous material plan (ONP) for products, the previous sub-assembly programme (SUBFILE) and sub-assembly queue (SUBQ) for sub-assemblies, and the supplier schedule (SCHEDULE) and material input file (INPUT) for components.

Products and sub-assemblies in allocation cause the reservation of components in IFILE. The initial stock status is contained within the item master file, which also defines a number of static parameters and relationships defining each product or component.

5. Experimental Control

The master STOCKMODEL program provides executive control over each simulation experiment. The simulation environment is first established by reference to the run parameters, initial conditions and sampling profiles, following which the start conditions may, optionally, be displayed.

The execution of the simulation experiment is by cycling through the three primary execution segments: QTREVENTS, MONTHEVENTS and WEEKEVENTS. The quarterly and monthly activities are by-passed if the appropriate control timers are not zero. The control timers (TQTR, TMMONTH) are reset by their dependent segments.

Following each cycle, the control timers are decremented, as is the timing field in each of the product queues. The week counter is incremented and the elapsed simulation time compared with the pre-defined run time. Completion of the simulation experiment causes an "end of run" message to be displayed and the run to be terminated.
Summarised Flow Chart- Master Program Stockmodel
PROGRAM(STKM)

COMpress INTEGER AND LOGICAL

INPUT 1=CRO

OUTPUT 2=LPO

END

MASTER STOCKMODEL

REAL AFILC(25,3), TREN(I,7), PRODMIX(I)

INTEGER TABLE1(7,2), TABLE2(7,2), TABLE3(7,2), TABLE4(7,2), TABLE5(7,2), TABLE6(7,2), TABLE7(7,2), TABLE8(7,2), TABLE11(2,2), IFILE(25,23), OCFMIN, OCFNOM, 2RESPONSE, EGBUF, PCBUF, CPRBUF, COMPRFR, RUNTIME, ORDHIST, 5(12,10,7)

S(10,10), SUBFILE(17,26), SCHEDULE(25,13), INPUT(25,52), OFILE(125,5)

475,10), ORDB(5,52,2075), TOTAL(5,52), COL(10,7), WEEK, REFWEK, EXFILE(25,52), REQFILE(25,52), ORDCOUNT, DELPLAN(5,52), ALLQ(12,50,6), LNEQ(5,50,5), TESTQ(5,50,5), EQUPERF(10,30), ITEMPERF(10,30), ORDPERF(72,30), NETPLAN(5,52), PIPELINE(20,6,4), DUE, QTY, TIME, ORDN, ITEMNO, CAP

8, LEADMTC(5,52), SUBQ(50,4), DIAGFILE(1072), ICOUNT(25), ISHORT(25)

DATA TABLE1(1,1)/=4/, TABLE1(2,1)/=-3/, TABLE1(3,1)/=-2/, TABLE1(4,1)/=1/, TABLE1(5,1)/=0/, TABLE1(6,1)/=1/, TABLE1(7,1)/=2/, TABLE1(1,2)/=1/, TABLE2(2,2)/=3/, TABLE2(3,2)/=9/, TABLE2(4,2)/=23/, TABLE2(5,2)/=63/, TABLE2(6,2)/=93/, TABLE2(7,2)/=100/, TABLE2(1,2)/=15/, TABLE2(2,2)/=30/, TABLE2(3,2)/=80/, TABLE2(4,2)/=95/, TABLE2(5,2)/=100/, TABLE2(6,2)/=100/, TABLE2(7,2)/=200/, TABLE3(1,2)/=1/, TABLE3(2,2)/=65/, TABLE3(3,2)/=85/, TABLE3(4,2)/=95/, TABLE3(5,2)/=100/, TABLE3(6,2)/=100/, TABLE3(7,2)/=100/, TABLE3(8,2)/=100/, TABLE3(9,2)/=100/, TABLE4(3,2)/=10/, TABLE4(4,2)/=30/, TABLE4(5,2)/=50/, TABLE4(6,2)/=70/, TABLE4(7,2)/=90/, TABLE4(8,2)/=100/, TABLE4(9,2)/=100/, TABLE5(1,1)/=1/, TABLE5(2,1)/=2/, TABLE5(3,1)/=3/, TABLE5(4,2)/=4/, TABLE5(5,2)/=5/, TABLE5(6,2)/=6/, TABLE5(7,2)/=7/, TABLE5(8,2)/=8/, TABLE5(9,2)/=9/, TABLE5(10,2)/=10/, TABLE5(11,2)/=11/, TABLE6(1,1)/=1/, TABLE6(2,1)/=2/, TABLE6(3,1)/=3/, TABLE6(4,2)/=4/, TABLE6(5,2)/=5/, TABLE6(6,2)/=6/, TABLE6(7,2)/=7/, TABLE6(8,2)/=8/, TABLE6(9,2)/=9/, TABLE6(10,2)/=10/, TABLE7(1,1)/=1/, TABLE7(2,1)/=2/, TABLE7(3,1)/=3/, TABLE7(4,2)/=4/, TABLE7(5,2)/=5/, TABLE7(6,2)/=6/, TABLE7(7,2)/=7/, TABLE7(8,2)/=8/, TABLE7(9,2)/=9/, TABLE7(10,2)/=10/, TABLE8(1,1)/=1/, TABLE8(2,1)/=2/, TABLE8(3,2)/=3/, TABLE8(4,2)/=4/, TABLE8(5,2)/=5/, TABLE8(6,2)/=6/, TABLE8(7,2)/=7/, TABLE8(8,2)/=8/, TABLE8(9,2)/=9/, TABLE8(10,2)/=10/, TABLE9(1,1)/=1/, TABLE9(2,1)/=2/, TABLE9(3,2)/=3/, TABLE9(4,2)/=4/, TABLE9(5,2)/=5/, TABLE9(6,2)/=6/, TABLE9(7,2)/=7/, TABLE9(8,2)/=8/, TABLE9(9,2)/=9/, TABLE9(10,2)/=10/

SET LEVEL OF DIAGNOStICS

CALL PNMFCREATE(AFILC, IFILE)

READ(1,110) (DIAGFILE(I,1), I=1,10), (DIAGFILE(I,2), I=1,10)

110 FORMAT(1015/1015)

WRITE(2,112) (DIAGFILE(I,1), I=1,10), (DIAGFILE(I,2), I=1,10)

112 FORMAT(1H1,6X,'DIAGNOSTIC PROFILE'//6X,'WEEK'//9X,'1014/6X,'DIAG. LE
VEL'//2X,'1014//')

IPRINT=DIAGFILE(172)

WRITE(2,111) IPRINT

111 FORMAT(1H0,6X,'DIAGNOSTIC LEVEL SELECTED = 1,1,1//6X,'DIAGNOSTIC CH
OICE IS:-1//6X,'1 DETAIL DIAGNOSTICS/WARNINGS'//6X,'2 TRANSACTION
2 DETAILS/WARNINGS'//6X,'3 WEEKLY SUMMARIES/WARNINGS'//6X,'4 MONTHL
3Y SUMMARIES/WARNINGS'//6X,'5 QUARTERLY SUMMARIES/WARNINGS'//6X,'6
4 RUN SUMMARY/INITIAL CONDITIONS'//6X,'7 RUN SUMMARY ONLY'/////

WEEK=0

REFWEEK=0

MONTH=3

ORDCOUNT=0

900 READ(1,910)MODE, ISORT

910 FORMAT(211)

IF(MODE.EQ.0) GO TO 2000

GO TO (920,940,960)/MODE

920 WRITE(2,930)

930 FORMAT(1H0,6X,'MODE SELECTED IS MAKE TO STOCK ONLY!')

GO TO 965

940 WRITE(2,950)

950 FORMAT(1H0,6X,'MODE SELECTED IS MIXED STOCK AND ORDER!')

GO TO 965
960 WRITE(2,970)
970 FORMAT(1HO,6X,'MODE SELECTED IS MAKE TO ORDER ONLY')
965 I=I+1+1
971 WRITE(2,973)
973 FORMAT(1HO,6X,'PRIORITY RULE IS FIRST-IN-FIRST OUT')
GO TO 979
975 WRITE(2,977)
977 FORMAT(1HO,6X,'PRIORITY RULE IS DUE DATE')
979 DO 980 I=1,5
IF(IFILE(I,1).GT.1) GO TO 980
IFILE(I,1)=0
980 CONTINUE
990 DO 10 I=1,7
TABLE2(I,1)=1
TABLE3(I,1)=1
TABLE5(I,1)=1
10 TABLE6(I,1)=1
AVGE4=5.0
CVARN4=0.3
AVGE9=0.04
CVARN9=0.3
AVGE10=6.0
CVARN10=0.3
AVGE12=8.0
CVARN12=0.3
DO 12 I=1,25
ISHORT(I)=0
IF(IPRINT.GT.6) GO TO 120
WRITE(2,15)
15 FORMAT(1HO,6X,'SAMPLING DISTRIBUTIONS:'//)
WRITE(2,20) (TABLE1(J,1),J=1,7), (TABLE1(J,2),J=1,7)
20 FORMAT(1HO,6X,'NORMAL INPUT'//6X,P110//6X,P110)
WRITE(2,230) (TABLE2(J,1),J=1,7), (TABLE2(J,2),J=1,7)
30 FORMAT(1HO,6X,'ARREARS INPUT'//6X,P110//6X,P110)
WRITE(2,240) (TABLE3(J,1),J=1,7), (TABLE3(J,2),J=1,7)
40 FORMAT(1HO,6X,'EXPEDITED INPUT'//6X,P110//6X,P110)
WRITE(2,250) (TABLE5(J,1),J=1,7), (TABLE5(J,2),J=1,7)
50 FORMAT(1HO,6X,'NUMBER OF ITEMS'//6X,P110//6X,P110)
WRITE(2,260) (TABLE6(J,1),J=1,7), (TABLE6(J,2),J=1,7)
60 FORMAT(1HO,6X,'PRODUCT'//6X,P110//6X,P110)
WRITE(2,270) (TABLE7(J,1),J=1,7), (TABLE7(J,2),J=1,7)
70 FORMAT(1HO,6X,'QUANTITY'//6X,P110//6X,P110)
WRITE(2,280) (TABLE8(J,1),J=1,2), (TABLE8(J,2),J=1,2)
80 FORMAT(1HO,6X,'PART SHIP = 1? NO PART SHIP = 2'//'6X,P110//'6X,P110)
WRITE(2,290) (TABLE11(J,1),J=1,2), (TABLE11(J,2),J=1,2)
90 FORMAT(1HO,6X,'REJECT INPUT'//6X,P210//6X,P210)
WRITE(2,300) AVGE4,CVARN4,AVGE9,CVARN9,AVGE10,CVARN10,AVGE12,CVARN12
100 FORMAT(1HO,6X,'NORMAL DISTRIBUTIONS'//30X,'AVERAGE'//6X,'CO-EFF.OF VARIANCE'//6X,'NO OF ORDERS'//10X,'F8.2,10X,'F8.2//'6X,'REJECT PERCENT'//8X,'F8.2,10X,'F8.2//'6X,'RESCHED INPUT'//8X,'F8.2,10X,'F8.2//'6X,'PICKING TIME'//10X,'F8.2,10X,'F8.2//'6X)
ESTABLISH RUN PARAMETERS
120 READ(1,130)OCFMIN,OCFOM,RESPONSE,EQUIBUFF,PCBUFF,COMPUF,UTIL,
1(TREND(J),J=1,7)
130 FORMAT(1,130)OCFMIN,OCFOM,RESPONSE,EQUIBUFF,PCBUFF,COMPUF,UTIL,
1(TREND(J),J=1,7)
WRITE(2,135) OCFMIN, OCFNOM, RESPONSE, EQUUBFR, PCDBUFR, COMPUFR, UTIL
135 FORMAT(1HO,6X,IRUN PARAMETERS'//6X,'MINIMUM ORDER BOOK (OCFMIN)'=1,'116//6X,'NOMINAL ORDER BOOK (OCFNUM)'=1,116//6X,'SCHEDULE RESPONSE TIME IN MONTHS (RESPONSE)'=1,116//6X,'BUFFER STOCK WEEKS FOR COMPONENTS'//112X,'COMPONENTS'/'4X,'CAPACITY UTILISATION = 1/F4.2)

144 IF(IPRINT.GT.6) GO TO 146
WRITE(2,145) (TREND(J),J=1,7)
145 FORMAT(1HO,6X,'TREND PARAMETERS'//5X,F5.2)
146 PRODMIX(1)=TABLE6(1,2)
DO 147 J=1,6
147 PRODMIX(J+1)=FLOAT(TABLE6(J+1,2)-TABLE6(J,2))
IF(IPRINT.GT.6) GO TO 137
WRITE(2,148) (PRODMIX(J),J=1,7)
148 FORMAT(1HO,6X,'PRODMIX'//10X,F7.2)
137 DO 170 I=1,25
LEVEL=FILE(I,11)
IF(LEVEL.EQ.0) GO TO 170
GO TO(140,150,155,160),LEVEL
140 IF(LEVEL(1,20).NE.EQUBUFR)
GO TO 170
150 IF(LEVEL(1,20).NE.PCDBUFR)
GO TO 170
155 IF(LEVEL(1,20).NE.COMPUFR)
GO TO 170
160 CONTINUE
READ(1,180) RUNTIME
180 FORMAT(15)

SELECT RANDOM NUMBER SEEDS
185 READ(1,410) ISEED1,1SEED2,1SEED3,1SEED4,1SEED5,1SEED6,1SEED7,1SEED8
11,ISEED9,ISEED10,1SEED11,1SEED12
410 FORMAT(19,(16),16)
WRITE(2,450) (I1,1=1,12), ISEED1,1SEED2,1SEED3,1SEED4,1SEED5,1SEED6,1
1SEED7,1SEED8,1SEED9,1SEED10,1SEED11,1SEED12
450 FORMAT(1HO,6X,'RANDOM NUMBER SEEDS'//6X,1217//6X,1217)

ESTABLISH INITIAL CONDITIONS
CALL INITODS(ORDHIST,TABLE5,TABLE6,TABLE7,ORDCOUNT,PIPELINE,ISEE
104,1SEED5,1SEED6,1SEED7,AVG4,CVARN4,FILE,TABLE8,ISEED8,IPRINT)
IF(IPRINT.GT.6) GO TO 200
WRITE(2,190)
190 FORMAT(1HO,6X,'INITIAL CONDITIONS'//))
200 READ(1,210) TYPE
210 FORMAT(1A8)
IF(IPRINT.GT.6) GO TO 217
WRITE(2,215) TYPE
215 FORMAT(1HO,6X,1A8/)
217 CALL COMPATYPE,BLINK,IT)
GO TO(300,220),IT
220 CALL TRANSLATE 3(AFILF,TYPE,N,200)
LEVEL=FILE(N,11)
GO TO(230,250,250,270),LEVEL
230 READ(1,240) (ONP(N,J),J=1,7), (R(N,J),J=1,10)
240 FORMAT(715/10I5)
IF(IPRINT.GT.6) GO TO 200
WRITE(2,245) (ORDHIST(N,J),J=1,12), (ONP(N,J),J=1,7), (R(N,J),J=1,10)
245 FORMAT(1HO,6X,'ORDER HISTORY'//16X,1215//6X,'PREVIOUS MATERIAL PLAN
1,7X,715/6X,'RECOMMENDED LINE RATES'//7X,1015//)
GO TO 200
250 READ(1,260)(SUBFILE(N,J),J=1,16)
260 FILE(N,18)=0
J=0
262 J=J+1
READ(1,263)ORDNO,ITEMNO,DUE;QTY,TIME
263 FORMAT(515)
   IF(QTY.EQ.0)GO TO 264
   OUTPUT=FLOAT(TIME)
   IMAX=1
   M=N
   CALL STKALL(IFILE,QTY,M,ALLQ;ORDNO,ITEMNO,DUE,OUTPUT;ICOUNT,ISHORT
   1;IPRINT;IMAX;AFILE,8264)
   IF(J.LE.50)GO TO 262
264 J=0
265 J=J+1
   IF(J.GT.50)GO TO 272
   IF(SUBQ(J,1).NE.0)GO TO 265
   READ(1,275)(SUBQ(J,K),K=2,4)
275 FORMAT(1615)
   IF(SUBQ(J,3).EQ.0)GO TO 266
   SUBQ(J,1)=N
   IF(IPRINT.GT.1)GO TO 269
   WRITE(2;268)AFILE(N,1),(SUBQ(J,K),K=2,4)
269 FILE(N,18)=FILE(N,18)+SUBQ(J,3)
   GO TO 265
272 IF(IPRINT.GT.6)GO TO 200
266 WRITE(2;267)SUBFILE(N,J),J=1,16)
276 FORMAT(1615)
   IF(IPRINT.GT.6)GO TO 200
   READ(1,280)(SCHEDULE(N,J),J=2,13),(INPUT(N,J),J=14,26)
280 FORMAT(12t5/13fS)
   IF(IPRINT.GT.6)GO TO 200
   WRITE(2;280)(SCHEDULE(N,J),J=1,13),(INPUT(N,J),J=14,26)
290 FORMAT(1615)
   CALL QPCREATE(OPFILE;ORDBK,TOTAL,ALLQ,LINEQ;TESTQ,ORDCOUNT,REFWEE
   1K,IFILE;ISEED4;ICOUNT,ISHORT;IPRINT,AFILE,SUBQ)
   IF(IPRINT.GT.6)GO TO 400
   CALL QPRRT(OPFILE,AFILE)
   CALL QDBKPRT(OPFILE;AFILE,IFILE,TOTAL,REFWEEK)
   CALL AQPRINT(ALLG,WEEK,AFILE,REFWEEK)
   CALL LPRINT(LINEQ,WEEK,AFILE,REFWEEK)
   CALL SQPRINT(SUBQ,WEEK,AFILE,REFWEEK)
   CALL TPRINT(TESTQ,WEEK,AFILE,REFWEEK)
   CALL PQMFPRINT(AFILF,IFILE)
   GO TO 200
300 CALL WKPRT(1COUNT,ISHORT;EQUPERF;ITEMPERF,ORDPERF,IFILE,AFILE;
   1REFWEEK;WEEK;LEADTM)
350 START SIMULATION RUN
400 TQTR=0.0
   TMONTH=0.0
500 IF(TQTR.GT.0.0)GO TO 600
   CALL WPRT(COUNT,ISHORT;EQUPERF;ITEMPERF,ORDPERF,IFILE,AFILE;
   1REFWEEK;WEEK;LEADTM)
40RDPERF,EQUBUFIR)
600 IF(MONTH.GT.0.0) GO TO 700
   CALL MONTHEVENT(AFIE,IFILE;REQFILE,SUBFILE,WEEL,ORDHIST,IPRINT,TH
   ONTH,MONTH,OPFILE;DELPLAN,NETPLAN,OCFMIN,ORDBK,ALLQ,TOTAL)
   IF(IPRINT.GT.4) GO TO 700
   CALL WEEKPRINT(ICOUNT,ISHORT,EQUPERF;ITEMPERF,ORDPERF,IFILE,AFI
   LEWEEK;WEEK,LEADTM)
700 CALL WEEKEVENT(AFIE,IFILE,Table3,Table5,Table7,Table9,Table11,AVG
   1E4,AVR44,AVG9,AVG10,ISEED5,ISEED4,ISEED5,ISEED6,ISEED7,IS
   2SEED7,ISEED8,ISEED9,ISEED10,ISEED11,ISEED12,Table8,AVG12,AVG
   N12,CVRN12,CVRN10,37ORDERCUT,PIPIINE,DELPLAN,OCFMIN,THEEK,TOTA
   L,ORDBK,OPFILE,ORDHIST,4INPUT,EXFILE;SCHEDULE,REQFILE,ALLO,LINEQ,TESTQ,SUBFIL
   E,EQUPEF,ITEMPERF,ORDPERF,NETPLAN,REWEEK,IPRINT,ICOUNT,ISHORT,CAP,LEADTM,ISOR
   6T,SUBQ,DIAGFILE,MODE,RTIME)
   WEEK=WEEK+1
   T0TR=T0TR-1.0
   TM0NTH=TM0NTH-1.0
   DO 800 I=1,12
      DO 800 J=1,50
         ALLQ(I,J,5)=ALLQ(I,J,5)+5
   800 CONTINUE
   DO 810 I=1,5
      DO 810 J=1,50
         TESTQ(I,J,5)=TESTQ(I,J,5)+5
         LINEQ(I,J,5)=LINEQ(I,J,5)+5
   810 CONTINUE
   DO 820 J=1,50
      SUBQ(J,4)=SUBQ(J,4)+5
   IF((REWEEK+WEcK)<RTIME) GO TO 500
   WRITE(2,1000)
1000 FORMAT(1HO,6X,IEND OF RUN!)
   GO TO 900
2000 STOP
END
Subroutine ALLOCATE is the order release mechanism which determines the quantity and timing of a factory order.

The process described demonstrates one of the basic differences between the procedure currently employed and a full M.R.P. approach. In M.R.P., order release is recommended by the system as planned orders become due, and the detail of each shop order in terms of quantity and timing is calculated to support the Master Production Schedule and ultimately, the customer order commitment. The present system does not support a customer order book nor, by definition, a Master Production Schedule which reflects the known customer order commitment in a dynamic manner. Customer orders are contained within a factory order book which is manually maintained, and supporting procedures are required to interpret the factory order book into time phased order releases, or "allocations".

The subroutine has been separated into two distinct sectors; the logic required to support sub-assembly manufacture, and the logic required to produce equipments.

a) Sub-assemblies

Each sub-assembly has a new programme re-generated at the time of the quarterly planning process, as defined in subroutine EBQ and SUBPROG. This weekly programme is reviewed each month to accommodate any deviation from plan which results in excess or lack of stock.

Since the weekly programme SUBFILE maintained by subroutine SUBPROG is considered to be always current, the requirement of subroutine ALLOCATE is to search for items which are due for order release and initiate order release action.

The sub-assembly programme in SUBFILE contains make quantities against the week that they are required to be completed and available in stock. Thus the subroutine must search for items with a due date equal to or less than the current week plus the manufacturing and component picking lead time. Items due
for release have the sub-assembly reference, quantity and due
date passed to subroutine STKALL for allocation to take place.

b) Equipment

The procedure for equipments differs substantially from that
for sub-assemblies. The requirement to allocate is not available
in a time-phased plan because of the inability to maintain
a factory order book within the system. The factory order book
must be scanned each week to search for customer orders which
are due for release, some of which may be recent additions to the
order book.

The segment for equipments is further sub-divided for ease of
processing into two parts; a segment for overdue products and a
segment for due products. The logic in each case is similar.

For each manufactured batch, the first requirement is to
determine the quantity and timing of the order release. The value
of work in progress is compared with the nominal queue (or lead
time) contained within IFILE. If the queue is less than the
nominal queue plus one week extended by the plan rate within
REQFILE an order release will be recommended. The quantity
recommended for release will be two weeks worth of requirement.

The next stage is to determine which customer orders are due for
allocation. The product order book contained within ORDBK is
searched by due date to identify a requirement which has not yet
been dispatched. The corresponding orders placed file record
contained within OPFILE is then retrieved to identify any
requirements which are still in "open" status.

Orders may be satisfied from two sources; semi-finished stock or
make to order. Semi-finished stock is held between the line
assembly and test activities and serves two purposes. Firstly,
to buffer the production lines against the unstable order input
patterns, and secondly to offer a shorter delivery lead time
than fully make to order.
The free stock for the required product is checked and, if the full item quantity required can be satisfied from stock, the allocated stock will be augmented by the required quantity and the OPFILE status changed from "unallocated" to "ex-stock".

If the item quantity cannot be satisfied from stock, the order will be wholly or partially satisfied from the next batch to be manufactured.

Orders eligible for manufacture are allocated in subroutine STKALL until the full allocation quantity recommended for order release has been taken up. Items thus allocated are moved from "open" to "on-line" status in OPFILE.

The order book is searched for customer requirements up to two weeks beyond the normal allocation date. If sufficient customer orders are not available within this period, a stock order is initiated to take up any balance in the allocation quantity. The stock order is assigned a negative order number to differentiate it from customer orders, since no ORDBK or OPFILE records will be created for stock requirements.
SUBROUTINE
ALLOCATE

SELECT NEXT
PRODUCT

AVAILABLE =
QUANTITY IN
ALLOCATION +
WIP

Sub-assembly

TYPE OF
PRODUCT?

Equipment

ESTABLISH
MINIMUM
TIME

SELECT NEXT
ORDE IN
ORDER BOOK

SUBROUTINE
STKALL

Overdue
Orders

SUBROUTINE
ALLOCATE
STOCK & MOVE TO
WIP QUEUE

INCREASE
ALLOCATED STOCK

SUBROUTINE
STKALL

PLAN QTY

SUBROUTINE
NORMAL

Y

TRANSIENT
ASSEMBLY?

ESTABLISH DUE
DATE

ESTABLISH PLAN QTY

QTY = 0?

Y

ESTABLISH MINIMUM
TIME

ALLOCATE
STOCK & MOVE TO
ALLOCATION QUEUE

N

SUBROUTINE
NORMAL

Y

ESTABLISH
PLAN QTY

AVAILABLE ≥ PLAN?

N

ESTABLISH
ALLOCATION QTY

Y

SUBROUTINE
NORMAL

ESTABLISH MINIMUM
TIME

SELECT NEXT
ORDER IN
ORDER BOOK

SUFFICIENT
EQUIPMENT
STOCK

N

SET STATUS
TO "ALLOCATE TO STOCK"

SUBROUTINE
STKALL

ALLOCATE
STOCK & MOVE TO
WIP QUEUE

Cont. over.
Select next order in order book

Y

Sufficient equipment stock

N

Allocate stock & move to WIP queue

Set status to "Allocate to Stock"

Increase allocated stock

Insufficient orders due?

Y

Create stock order

Allocate stock & move to WIP queue

N

Due Orders

Summarised Flow Chart - Subroutine Allocate
ALLOCATE

SUBROUTINE ALLOCATE(FILE, REQFILE, ORDNO, STKORD, WEEK, OPFILE, ISEED12
  I; ALLQ, ICOUNT, ISHORT, SUBFILE, AVG12, CVARN12, REFWEEK, IPRINT, AFIL
  DGFILE, MODE, RUNTIME, LINEQ)
REAL AFIL(25, 3)
INTEGER WEEK, REQFILE(25, 52), FILE(25, 23), WIP, ALLQTY, STKORD(20), Z, O
  1DBK( 5, 52, 20, 5), OPFILE(125, 5, 5, 10), ORD, ALLQ(12, 50, 6), QTY, SUBFILE(2
  12, 25), DDATE, REFWEEK, DIAGFILE(10, 2), RUNTIME, ICOUNT(25), ISHORT(25)
3LINIQ(5, 50, 5)
IF(IPRINT.GT.1)GO TO 1
WRITE(2, 400)
400 FORMAT(1HO, 6X, 1** ** ** ALLOCATE ** ** ** **//)
1 DO 200 I=1, 12
  IAVAIL=0
  ISTK=0
  DO 4 K=1, 52
    IF(ALLOQ(I, K, 1).GE.0)GO TO 5
    ISTK=ISTK+ALLOQ(I, K, 4)
5 IF(LINEQ(I, K, 1).GE.0)GO TO 4
    ISTK=ISTK+LINEQ(I, K, 4)
4 IAVAIL = IAVAIL + ALLOQ(I, K, 4)
    IAVAIL = IAVAIL + FILE(I, 18)
    IF(FILE(I, 11)
      GO TO (872, 27200)
    IF
    SEGMENT FOR SUB-ASSEMBLIES
2 ORDNO=0
    ITEMNO=0
    IF(FILE(I, 15).EQ.1)GO TO 200
    IDATE=WEEK+FILE(I, 12)+1+REFWEEK
    QTY=SUBFILE(I, WEEK)
    IF(QTY.EQ.0)GO TO 200
    IMAX=0
    OUTPUT=8.0
    IF(ISEED12.EQ.0)GO TO 7
    CALL NORMAL(AVG12, OUTPUT, ISEED12, CVARN12)
    IF(OUTPUT.GE.0.0)GO TO 7
    OUTPUT=0.0
7 CALL STKALI(FILE; QTY; I, ALLQ; ORDNO, ITEMNO, IDATE, OUTPUT; ICOUNT, ISH
    1ORT, IPRINT, IMAX, AFIL; 2200)
      GO TO 200

SEGMENT FOR EQUIPMENTS
8 ALLQTY=0
    IPLAN=0
    CHECK FOR HIGH W.I.P. * IN ALLOCATION
    M=WEEK+FILE(I, 12)+2
    DO 10 J=WEEK+M
10 IPLAN=IPLAN+REQFILE(I, J)
    IF(IAVAIL.GE.IPLAN)GO TO 200

ESTABLISH ALLOCATION QUANTITY, ORDER NUMBER & TIME
    IMAX=1
    DO 15 M=1, 50
      IF(ALLOQ(I, M, 6).LE.IMAX)GO TO 15
      IMAX=ALLOQ(I, M, 6)

15 CONTINUE
   IMAX = IMAX + 1
   OUTPUT = 8.0
   IF (ISEED12.EQ.0) GO TO 19
   CALL NORMAVG12, OUTPUT, ISEED12, CVARN12
   IF (OUTPUT.GE.0.0) GO TO 19
   OUTPUT = 0.0
19 DO 20 M = WEEK + 2, WEEK + 3
20 ALLQTY = ALLQTY + REQFILE(I, M)
   IF (IPRINT.GT.2) GO TO 21
   WRITE(2, 330) AFFILE(I, 1), ALLQTY, IMAX, OUTPUT
   CONTINUE
   GO TO 33
30 FORMAT(1HO, 6X, 'PRODUCT =', A8, 2X, 'ALLOCATION QTY =', 16, 2X, 'SHOP ORDER =', 16, 2X, 'TIMER =', F6.1)
33 N = ORDNO
   M = ITEHNO
   J = QTY
   K = WEEK DUE
21 IF (ALLQTY.LE.0) GO TO 200

SEGMENT FOR OVERDUE ITEMS

 DO 100 K = 1, 52
 DO 100 L = 1, 20
 IF (ORDBK(I, K, L, 4) .EQ. 0) GO TO 100
 DATE = K - 52
 IF (ORDBK(I, K, L, 3) .EQ. 0) GO TO 100
 N = ORDBK(I, K, L, 1)
 M = ORDBK(I, K, L, 2)
 J = ORDBK(I, K, L, 3)
 DO 25 KEVNO = ', 125
 IF (OPFILE(KEVNO, M, JJ, 3) .GT. WEEK) GO TO 150
25 CONTINUE
 GO TO 33
27 DO 30 JJ = 1, 5
 IF (OPFILE(KEVNO, M, JJ, 3) .GT. DATE) GO TO 40
30 CONTINUE
33 IF (IPRINT.GT.5) GO TO 100
 WRITE(2, 350) N, M, DATE, OPFILE(KEYNO, M, JJ, 3), REFWEEK
35 FORMAT(1HO, 6X, 'INCOMPATIBLE DATA IN OPFILE FOR ORDER NO. ', 14, ' ITEN NO. ', I4, ' DUE WEEK ', 14, 6X, 'ACTUAL WEEK ', 14, 2X, 'REFWEEK ', 14)
 IF (IPRINT = 1)
 DO 37 IZ = 1, 10
37 DIAGFILE(IZ, 2) = 1
 RUNTIME = REFWEEK + WEEK + 4
 GO TO 100
40 IF (OPFILE(KEYNO, M, JJ, 4), .EQ. 0) GO TO 100
 IF (OPFILE(KEYNO, M, JJ, 4), GT, (IFILE(I, 16) + ISTK = IFILE(I, 17))) GO TO 150

TAKE FROM STOCK

 IFILE(I, 17) = IFILE(I, 17) + OPFILE(KEYNO, M, JJ, 4)
 OPFILE(KEYNO, M, JJ, 5) = OPFILE(KEYNO, M, JJ, 5) + OPFILE(KEYNO, M, JJ, 4)
 OPFILE(KEYNO, M, JJ, 4) = 0
 GO TO 100

MAKE ON LINE

50 IF (MODE .EQ. 1) GO TO 100
 IF (ALLQTY.LT.0) OPFILE(KEYNO, M, JJ, 4) GO TO 150
 QTY = OPFILE(KEYNO, M, JJ, 4)
 IDATE = OPFILE(KEYNO, M, JJ, 3)
CALL STKALL(IFILE,QTY,1,ALLQ,N,M,IDATE,OUTPUT,ICOUNT,ISHORT,IPRINT
IT,IMAX,AFILE,'8100)
OPFILE(KEYNO,M,JJ,6)=OPFILE(KEYNO,M,JJ,6)+QTY
OPFILE(KEYNO,M,JJ,4)=0
ALLQTY=ALLQTY-QTY
100 CONTINUE

SEGMENT FOR DUE ITEMS

DO 142 K=1,WEEK+13
DO 140 L=1,20
DBATE=K
IF(ORDDBK(I,K,L,4),EQ.1)GO TO 140
IF(ORDDBK(I,K,L,3),EQ.0)GO TO 140
N=ORDDBK(I,K,L,1)
M=ORDDBK(I,K,L,2)
J=ORDDBK(I,K,L,3)
DO 110 KEYNO=1,125
IF(OPFILE(KEYNO,1,I,1),EQ.H)GO TO 115
110 CONTINUE
GO TO 125
115 DO 120 JJ=1,5
IF((OPFILE(KEYNO,M,JJ,3)=REFWEEK),EQ.K)GO TO 130
120 CONTINUE
125 IF(IPRINT.GT.5)GO TO 140
WRITE(2,35)N7M,DDATE,OPFILE(KEYNO,M,JJ,3),REFWEEK
IPRINT=1
DO 127 IZ=1,10
127 DIAGFILE(IZ,2)=1
RUNTIME=REFWEEK+WEKD+4
GO TO 140
130 IF(OPFILE(KEYNO,M,JJ,4),EQ.0)GO TO 140
IF(OPFILE(KEYNO,M,JJ,4),GT,(IFILE,I,16)+ISTK=IFILE(I,17))GO TO 1135
135 CONTINUE
IF(MODE.EQ.1)GO TO 140
IF(ALLQTY.LE.OPFILE(KEYNO,M,JJ,4))GO TO 150
QTY=OPFILE(KEYNO,M,JJ,4)
IDATE=OPFILE(KEYNO,M,JJ,3)
CALL STKALL(IFILE,QTY,1,ALLQ,N,M,IDATE,OUTPUT,ICOUNT,ISHORT,IPRINT
IT,IMAX,AFILE,'8140)
OPFILE(KEYNO,M,JJ,6)=OPFILE(KEYNO,M,JJ,6)+QTY
OPFILE(KEYNO,M,JJ,4)=0
ALLQTY=ALLQTY-QTY
IF(MODE.EQ.3)GO TO 140
140 CONTINUE
IF(K,GE,WEEK+6)GO TO 145
145 CONTINUE
142 CONTINUE
IF(MODE.EQ.3)GO TO 200
145 STKORD(I)=STKORD(I)+1
H=1
N=STKORD(I)
IDATE=WEEK+4+REFWEEK
NEGATIVE ORDER NUMBER DENOTES STOCK ORDER

GO TO 180

150 IDATE=OPFILE(KEYNO,M,JJ,3)

180 CALL STKALL(IFILE;ALLQTY;I,ALLQ,N,M,IDATE,OUTPUT,ICOUNT,ISHORT,IPRINT,IMAX,AFILE,&200)

IF(N.LE.0)GO TO 200

OPFILE(KEYNO,M,JJ,4)=OPFILE(KEYNO,M,JJ,4)-ALLQTY

OPFILE(KEYNO,M,JJ,6)=OPFILE(KEYNO,M,JJ,6)+ALLQTY

200 CONTINUE

IF(IPRINT.GT.3)GO TO 500

CALL AQPRINT(ALLQ,WEEK,AFILE,REFWEEK)

500 RETURN

END
Four print routines are available as diagnostic aids to report the contents of the queues: ALLQ, LINEQ, TESTQ, SUBQ.

Each queue may be reported at any time, and is identified by the week number provided by the calling segment.

Details of each queue are presented in product sequence (except for SQPRINT), but only for each product with a finite queue content. SQPRINT presents the contents of the sub-assembly queue in the sequence defined in SUBQ. Summary values indicating the quantity in each independent queue are displayed for ease of analysis.
 Summarised Flow Chart - Subroutine Aqprint
SUBROUTINE AQPRINT(ALLQ, WEEK, AFILE, REFWEEK)
REAL AFILE(25, 3)
INTEGER ALLQ(12, 50, 6), WEEK, REFWEEK
IDATE=WEEK+REFWEEK
WRITE(2; 10) IDATE
10 FORMAT(140, 'ITEMS IN ALLOCATION FILE FOR WEEK, 16//10X, 'ORDER
1NO., '5X; 'ITEM NO., '5X; 'DUE DATE', '5X, 'QUANTITY', '8X, 'TIMER', '5X,
2!SHOP ORDER!/
DO 200 I=1, 12
IQTY=0
IF(ALLQ(I, 1, 4).EQ.0) GO TO 200
WRITE(2; 20) AFILE(I, 1)
20 FORMAT(1H0, 6X, A8)
DO 100 J=1, 50
IF(ALLQ(I, J, 4).EQ.0) GO TO 100
IQTY=IQTY+ALLQ(I, J, 4)
WRITE(2; 30) (ALLQ(I, J, K), K=1, 6)
30 FORMAT(1H5X, 6X, 16X)
100 CONTINUE
WRITE(2; 50) IQTY
150 FORMAT(1H5, 90X, 'TOTAL =', I6)
200 CONTINUE
RETURN
END

<table>
<thead>
<tr>
<th>ORDERNO.</th>
<th>ITEM NO.</th>
<th>DUE DATE</th>
<th>QUANTITY</th>
<th>TIMER</th>
<th>SHOP ORDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF6AM 01</td>
<td>-2</td>
<td>21</td>
<td>86</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>1</td>
<td>50</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>4</td>
<td>25</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>-3</td>
<td>1</td>
<td>111</td>
<td>9</td>
<td>5</td>
</tr>
</tbody>
</table>

| MF6AM 02 | -3       | 1        | 73       | -3    | 4          |
|          | 24       | 2        | 7        | 3     | 4          |
|          | 24       | 3        | 33       | 9     | 5          |

| MF6AM 03 | -4       | 1        | 46       | 2     | 5          |

| AT12345  | 0        | 0        | 21       | 160   | -2         | 0          | TOTAL = 160 |
| AT12801  | 0        | 0        | 22       | 100   | 1          | 0          | TOTAL = 100 |
| AT12802  | 0        | 0        | 22       | 92    | -2         | 0          | TOTAL = 184 |

Sample Allocation File
SUBROUTINE LQPRINT(LINEQ, WEEK, AFILE, REFWEEK)
REAL AFILE(25, 3)
INTEGER LINEQ(5, 50, 5), WEEK, REFWEEK
Iodate = WEEK * REFWEEK
WRITE(2, 10) DATE
10 FORMAT(1HO, /6X, 'ITEMS IN LINE QUEUE (WIP) FOR WEEK', '16', /10X, 'ORDER NO.', '5X', 'ITEM NO.', '175X', 'DUE DATE', '5X', 'QUANTITY', '5X', 'TIMER' //)
DO 200 1 = 1, 5
IQTY = 0
IF (LINEQ(1, I), EQ, 0) GO TO 200
WRITE(2, 20) AFILE(1, 1)
20 FORMAT(1HO, 6X, A8)
DO 100 J = 1, 50
IF (LINEQ(1, J), EQ, 0) GO TO 100
IQTY = IQTY + LINEQ(1, J)
WRITE(2, 30) (LINEQ(I, J), K = 1, 5)
30 FORMAT(14H$, 5X, 5(7X, 16))
100 CONTINUE
WRITE(2, 150) IQTY
150 FORMAT(1H$, 80X, 'TOTAL = ', 16)
200 CONTINUE
RETURN
END

ITEMS IN LINE QUEUE (WIP) FOR WEEK 19

<table>
<thead>
<tr>
<th>ORDER NO.</th>
<th>ITEM NO.</th>
<th>DUE DATE</th>
<th>QUANTITY</th>
<th>TIMER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF6AM 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>19</td>
<td>20</td>
<td>-12</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>19</td>
<td>87</td>
<td>-7</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>120</td>
<td>56</td>
<td>-12</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>121</td>
<td>86</td>
<td>-3</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>27</td>
<td>10</td>
<td>+7</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>28</td>
<td>50</td>
<td>+3</td>
</tr>
<tr>
<td>TOTAL =</td>
<td>328</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF6AM 02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>19</td>
<td>10</td>
<td>-12</td>
</tr>
<tr>
<td>-2</td>
<td>1</td>
<td>19</td>
<td>78</td>
<td>-7</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>20</td>
<td>25</td>
<td>-12</td>
</tr>
<tr>
<td>-3</td>
<td>1</td>
<td>21</td>
<td>73</td>
<td>-3</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>25</td>
<td>2</td>
<td>-7</td>
</tr>
<tr>
<td>24</td>
<td>2</td>
<td>28</td>
<td>7</td>
<td>+3</td>
</tr>
<tr>
<td>TOTAL =</td>
<td>195</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF6AM 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>1</td>
<td>19</td>
<td>32</td>
<td>-7</td>
</tr>
<tr>
<td>-3</td>
<td>1</td>
<td>20</td>
<td>46</td>
<td>-2</td>
</tr>
<tr>
<td>TOTAL =</td>
<td>78</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC15PU</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>18</td>
<td>18</td>
<td>-7</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>19</td>
<td>4</td>
<td>-7</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>19</td>
<td>16</td>
<td>-7</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>20</td>
<td>10</td>
<td>-2</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>21</td>
<td>8</td>
<td>-7</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>12</td>
<td>-2</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>27</td>
<td>22</td>
<td>-2</td>
</tr>
<tr>
<td>TOTAL =</td>
<td>110</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sample W.I.P. File
SUBROUTINE TOPRINT(TESTQ, WEEK, AFILE, REFWEEK)
REAL AFILE(25, 3)
INTEGER TESTQ(5, 50, 5), WEEK, REFWEEK
IDATE=WEEK+REFWEEK
WRITE(2,10) IDATE
10 FORMAT(1HO /6X, 'ITEMS IN TEST QUEUE (WIP) FOR WEEK', 16//10X, 'ORDER NO. ', 5X, 'ITEM NO. ', 5X, 'DUE DATE ', 5X, 'QUANTITY ', 5X, 'TIMER'/)
DO 200 I=1, 5
IQTY=0
IF(TESTQ(I, 1, 4).EQ.0)GO TO 200
WRITE(2,20) AFILE(I, 1)
20 FORMAT(1HO, 6X, A8)
DO 100 J=1, 50
IF(TESTQ(I, J, 4).EQ.0)GO TO 100
IQTY=IQTY+TESTQ(I, J, 4)
WRITE(2,30)(TESTQ(I, J, K), K=1, 5)
30 FORMAT(1H , 5X, 5(7X, 16))
100 CONTINUE
WRITE(2,150) IQTY
150 FORMAT(1H+, 80X, 'TOTAL = '1, 16)
200 CONTINUE
RETURN
END

ITEMS IN TEST QUEUE (WIP) FOR WEEK 19

<table>
<thead>
<tr>
<th>ORDER NO.</th>
<th>ITEM NO.</th>
<th>DUE DATE</th>
<th>QUANTITY</th>
<th>TIMER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF6AM 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>17</td>
<td>20</td>
<td>+12</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>17</td>
<td>25</td>
<td>+12</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>17</td>
<td>25</td>
<td>-7</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>18</td>
<td>19</td>
<td>-12</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>18</td>
<td>43</td>
<td>-7</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>18</td>
<td>13</td>
<td>-2</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>19</td>
<td>55</td>
<td>-2</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>19</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>TOTAL = 220</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| MF6AM 02  |          |          |          |       |
| 5         | 4        | 17       | 10       | -12   |
| 14        | 1        | 18       | 15       | -12   |
| 14        | 1        | 18       | 10       | -7    |
| 14        | 1        | 18       | 10       | -7    |
| 15        | 1        | 28       | 10       | -2    |
| 15        | 1        | 19       | 10       | 3     |
| TOTAL = 90 |

| MF6AM 03  |          |          |          |       |
| 6         | 1        | 16       | 8        | -22   |
| 1         | 16       | 8        | -17     |
| 9         | 1        | 17       | 9        | -17   |
| 13        | 2        | 18       | 5        | -12   |
| 13        | 3        | 18       | 4        | -12   |
| 13        | 3        | 18       | 1        | -7    |
| 13        | 4        | 18       | 5        | -7    |
| TOTAL = 60 |

| ACT5PU   |          |          |          |       |
| 11        | 2        | 17       | 20       | -2    |
| 11        | 2        | 18       | 2        | -2    |
| 11        | 2        | 18       | 18       | 3     |
| 15        | 2        | 19       | 4        | 3     |
| TOTAL = 44 |

Sample Test W.I.P. File
SUBROUTINE SQPRINT(SUBQ,WEEK,AFILE,REFWEEK)
REAL AFILF(25,3)
INTEGER WEEK,SUBQ(50,4),REFWEEK
IDATE=WEEK+REFWEEK
WRITE(2,10) IDATE
10 FORMAT(1H0/6X,'ITEMS IN SUB-ASSEMBLY W.I.P. FOR WEEK',/16/10X,
1'PART NO.',5X,'DUE',5X,'QUANTITY',5X,'TIMER'/)
IQTY=0
DO 100 J=1,SO
IF(SUBQ(J,3).EQ.0)GO TO 100
IQTY=IQTY+SUBQ(J,3)
IPROD=SUBQ(J,1)
WRITE(2,50) AFILE(IPROD,1),(SUBQ(J,N),N=2,4)
50 FORMAT(1H0/10X,A8,5X,I4,6X,14.8X,14)
100 CONTINUE
WRITE(2,150) IQTY
150 FORMAT(1H0/60X,'TOTAL = ',I6)
RETURN
END

ITEMS IN SUB-ASSEMBLY W.I.P. FOR WEEK

<table>
<thead>
<tr>
<th>PART NO.</th>
<th>DUE</th>
<th>QUANTITY</th>
<th>TIMER</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT12801</td>
<td>18</td>
<td>62</td>
<td>-12</td>
</tr>
<tr>
<td>AT12802</td>
<td>18</td>
<td>92</td>
<td>-12</td>
</tr>
<tr>
<td>AT12345</td>
<td>18</td>
<td>160</td>
<td>-7</td>
</tr>
<tr>
<td>AT12345</td>
<td>19</td>
<td>160</td>
<td>-2</td>
</tr>
<tr>
<td>AT27396</td>
<td>20</td>
<td>122</td>
<td>-12</td>
</tr>
<tr>
<td>AT12345</td>
<td>20</td>
<td>160</td>
<td>-2</td>
</tr>
<tr>
<td>AT12345</td>
<td>20</td>
<td>100</td>
<td>-2</td>
</tr>
<tr>
<td>AT12345</td>
<td>21</td>
<td>160</td>
<td>3</td>
</tr>
<tr>
<td>AT12802</td>
<td>21</td>
<td>92</td>
<td>3</td>
</tr>
</tbody>
</table>

TOTAL = 1108

Sample Sub-assembly Queue
The receipt of purchased items is simulated in the subroutine ARRIVAL. The pattern of receipts for each part number has been derived from the supplier schedule in subroutine RECEIVE and contained in the file named INPUT.

The week number is available to the subroutine, thus the expected receipt for each component for the appropriate week can be derived from file INPUT.

Subroutine ARRIVAL spans the events from receipt at the receiving bay to location in stock. The facility to audit quality is provided within the logic, but the quantity received complies with the data specified in file INPUT.

The quality audit logic follows the process outlined below:

a) A histogram is sampled to determine the presence of poor quality. Items which pass this test will proceed to the stock update phase.

b) Items which fail will have a second sample taken to determine the size of the reject quantity. If this is less than 5%, the reject items are returned to the supplier. The remainder will update the stock file. Reject quantities of greater than 5% cause rejection of the complete batch.

c) Quantities which have been rejected are rescheduled in file INPUT by sampling from a histogram defining the number of weeks required to recycle the batch.

A finite receipt quantity will cause two actions:

- update of the physical stock field in IFILE
- adjustment of the supplier schedule.
Receipts are assumed to downdate the file SCHEDULE in chronological sequence, starting with any arrears where applicable. Maintenance of the supplier schedule status is important to the correct functioning of the quarterly material planning logic and the relative stability of the supplier commitment.

The sampling profiles employed by subroutine ARRIVAL are:

(i) Probability that a reject batch has arrived. Histogram contained within TABLE 11.

<table>
<thead>
<tr>
<th>Value</th>
<th>Relative frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Fail)</td>
<td>20</td>
</tr>
<tr>
<td>2 (Pass)</td>
<td>80</td>
</tr>
</tbody>
</table>

(ii) Proportion of batch rejected.
Normal distribution as defined by AVGE9, CVARN9
(eg. average = 0.04; coefficient of variation = 0.3)

(iii) Number of weeks rescheduled.
Normal distribution as defined by AVGE10, CVARN10
(eg. average = 4.0; coefficient of variation = 0.3)
SUBROUTINE ARRIVAL

Start

Select next Component

Sample for Rejects

Reject present?

Y

Sample for Reject Proportion

Reject G.T. 5%?

N

Reject whole Batch

Reject Failed Items

Y

Sample Reschedule Date

Reschedule Fail Quantity

Accept Pass Quantity into Stock

Delog Schedule and Input Files

Last Item?

Y

Stop

N

Reject Histogram

Reject Proportion

Reschedule Period

Summarised Flow Chart - Sub-routine Arrival
ARRIVAL

SUBROUTINE ARRIVAL(SCHEDULE, INPUT, IFILE, WEEK, IPRINT, AVG9, CVAR9, A
VGE10, CVAR10, ISEED9, ISEED10, ISEED11, TABLE11, AFILE)
REAL AFILE
INTEGER SCHEDULE(25,13), INPUT(25,52), IFILE(25,23), WEEK, TABLE11(2,2)

1, REJECT
IF(IPRINT.GT.1) GO TO 350
WRITE(2,400)
400 FORMAT(1HO,6X,'** ARRIVAL ** ** */*
350 DO 100 J=1,25
IF(IFILE(J,11).NE.4) GO TO 100
IF(INPUT(J, WEEK).EQ.0) GO TO 100
SAMPLE FOR REJECTS......PASS=2 FAIL=1
IF(ISEED11.EQ.0) GO TO 8
CALL HSAMPLE 2(ISEED11, NUMBER, TABLE11)
GO TO (1,8)
Determine reject proportion:.......Greater than 5 PC rejects full batch

1 IF(IPRINT.GT.2) GO TO 2
WRITE(2,410)
410 FORMAT(1HO,6X,'BATCH FAILED AT INSPECTION!')
2 CALL NORMAL(AVG10, OUTPUT, ISEED9, CVAR9)
IF(OUTPUT.LT.0.05) GO TO 3
REJECT=INPUT(J, WEEK)
GO TO 4
3 REJECT=NINT(FLOAT(INPUT(J, WEEK)*OUTPUT)
4 IF(IPRINT.GT.2) GO TO 5
WRITE(2,500) OUTPUT, REJECT
500 FORMAT(1HO,6X,'OUTPUT=', F6.4,6X,'REJECT=', I6)
5 CALL NORMAL(AVG10, X, ISEED10, CVAR10)
N=NINT(X)
INPUT(J, WEEK+N)=INPUT(J, WEEK+N)+REJECT
INPUT(J, WEEK)=INPUT(J, WEEK)-REJECT
IF(IPRINT.GT.2) GO TO 9
WRITE(2,77) REJECT, N, INPUT(J, WEEK)
7 FORMAT(1HO,6X,'BATCH OF', I5, 'UNITS RESCHEDULED BY', I4, 'WEEKS
1,' 'LEAVING', I5, 'UNITS IN THE CURRENT PERIOD')
GO TO 9
8 IF(IPRINT.GT.2) GO TO 9
WRITE(2,420)
420 FORMAT(1HO,6X,'BATCH PASSED AT INSPECTION!')
9 IFILE(J,16)=IFILE(J,16)+INPUT(J, WEEK)
IF(IPRINT.GT.2) GO TO 10
WRITE(2,720) AFILE(J,1), IFILE(J,16), INPUT(J, WEEK), WEEK
200 FORMAT(1HO,6X,'PRODUCT=', 2X, A8, ' STOCK=', I5, ' INPUT=', I5, ' WEEK=', I5)
10 DO 50 K=1,13
IF(INPUT(J, WEEK).GE.SCHEDULE(J,K)) GO TO 20
SCHEDULE(J, K)=SCHEDULE(J, K)+INPUT(J, WEEK)
INPUT(J, WEEK)=0
GO TO 20
20 INPUT(J, WEEK)=INPUT(J, WEEK)-SCHEDULE(J, K)
SCHEDULE(J, K)=0
50 CONTINUE
60 IF(IPRINT.GT.3) GO TO 100
WRITE(2,330) (SCHEDULE(J,K), K=1,13)
300 FORMAT(1HO,6X,'REVISED SCHEDULE///6X,'1315)
WRITE(2,600) INPUT(J, K), K=1,720
600 FORMAT(1HO,6X,'REVISED INPUT///6X,'2015///')
100 CONTINUE
RETURN
END
**BATCH FAILED AT INSPECTION**

```
OUTPUT = 0.0239  REJECT# 2
BATCH OF 2 UNITS RESCHEDULED BY 6 WEEKS, LEAVING 98 UNITS IN THE CURRENT PERIOD
PRODUCT PN656043  STOCK 1348  INPUT 98 WEEK 2
REVISED SCHEDULE
  2 450 450 0 0 49 440 440 440 330 110 0 450
REVISED INPUT
  0 0 0 450 0 0 450 2 0 0 0 0 0 0 0 0 0 49 0
```

**BATCH PASSED AT INSPECTION**

```
PRODUCT PN69746  STOCK 1670  INPUT 450 WEEK 2
REVISED SCHEDULE
  0 450 450 493 660 760 760 790 880 880 880 450
REVISED INPUT
  0 0 0 0 0 450 0 0 493 450 0 0 660 0 0 0 760 0 760
```

Sample Receiving Report
Subroutine DELPERF extracts the base data for delivery performance to be calculated and summarises the statistics in histogram form.

Delivery performance statistics are held at three levels; equipment, item and order. Each delivery transacted in subroutine DESPATCH will cause a delivery performance assessment to be made. The data required for delivery performance calculation is the quantity despatched and the delay in weeks compared with the original due date. The despatch point is taken as the movement from test to the commercial warehouse, or despatch department. Negative delays represent early deliveries.

Each histogram will accommodate delays of up to 19 weeks and early deliveries of up to 8 weeks in weekly increments. Deviations in excess of these limits are held in two further categories; 20 weeks or more overdue and 9 weeks or more early.

a) Equipments

Equipment level records are held separately for each product to enable comparison to be made between products subject to different influences (e.g., order mix trends, stock policies, lead times). As each delivery is made, the appropriate delay interval is augmented by the quantity of equipments despatched.

b) Items

The item level record represents the performance achieved in despatching a complete order line against each due date. Thus, an item containing a quantity of equipments phased over five weeks will be analysed as five independent deliveries. Each completed delivery will increment the interval representing the final delivery by a quantity of one. Item level records are held for each equipment type.
c) Orders

Order statistics are maintained separately for "part shipment" and "no part shipment" orders. For "part shipment" orders, a completed delivery is acknowledged when all of the order line items for a particular week due have been despatched. "No part shipment" orders will not be acknowledged as despatched until the final delivery batch regardless of delivery phasing.

The order performance logic performs a second important function. When the delivery criteria have been met and a delivery initiated, the equipments will be removed from "in despatch" status in the orders placed file (OPFILE) signifying shipment to the customer. Until the final delivery against an order (or delivery week in the case of "part shipment") has been made, the equipments already despatched will remain in stock in the finished goods warehouse.

The magnitude of this "marshalled" stock is a function of the delivery performance and the degree of part shipment permissible.

The subroutine DELPERF is activated each time a delivery is registered in subroutine DESPATCH. The delay is determined by comparing the clock week against the week due held in the order book (ORDBK), allowing for overdue orders which have been relocated in ORDBK for reasons of file economy. The delay is truncated to the range -20 to +9 inclusive and modified to index a file with a range of 1 to 30.

The equipment performance is registered by augmenting the file EQUIPERF for the appropriate equipment type and calculated delay by the quantity despatched. Each quantity delivered will initiate a check against the orders placed file (OPFILE).

If the quantity in "despatch" status is equal to the total item quantity for the delivery week, the histogram representing item performance, ITEMPERF, will have the interval for the appropriate equipment and delay incremented by a quantity of one. Since the performance is measured by order line item and week due, the delay is identical to that used to determine equipment performance.
Failure to achieve a complete item delivery will result in an exit from the subroutine. A completed item will move into the order performance segments.

Orders which have been designated "part shipment permissible" may be shipped when all items promised against a common due date have been moved to despatch. In practice there are many different rules for part shipment, ranging from monthly deliveries, or to meeting certain transport line schedules.

The essence of part shipment is whether the customer will pay for the goods he receives, which is often related to the usefulness of the consignment. Consignment and subsequent invoicing with little chance of receiving payment moves the financial burden from commercial stock (valued at factory costs) to debtors (at sales invoice value). The compromise adopted by the model, of shipping complete weeks worth of equipment across all order lines, is reasonable in terms of customer acceptability.

The "part shipment" performance is established by searching OPFILE for all order lines with the same due date as the last despatched quantity. If all items with the common due date are in despatch, a shipment may be made and the performance statistics incremented accordingly. The delay is again identical to that used to determine equipment and item performance.

"No part shipment" performance follows a similar logic to "part shipment" performance, except that the full order is checked for "despatch" status. If the order is completed the delay is then calculated, being the difference between the calendar week at the longest due date on the order.

For "part shipment" orders, a shipment will cause the ORDBK record for the constituent products to be cleared. Shipment of a "no part shipment" order, and the final shipment of a part shipment order, will cause both the applicable ORDBK and OPFILE records to be cleared.
Summarised Flow Chart - Sub-routine Delperf
SUBROUTINE DELPERF(ORDBK, OPFILE, PROD, KEYNO, ITEM, DUE, WEEK, M, N, IPRINT, EQUPERF, ITEMPERF, ORDPERF, QTY)
INTEGER ORD BK(5, 5, 20, 5), OPFILE(125, 5, 5, 10), PROD, ORDNO, ITEM, DUE, WEEK, EQUPERF(10, 30), ITEMPERF(10, 30), ORDPERF(2, 30), DELAY, QTY
IF(ISPRINT.GT.1)GO TO 5
WRITE(2, 7)
5 ORDNO = ORDBK(PROD, DUE, N, 1)
L = DUE
Determine Due Date
IF(ORDBK(PROD, DUE, N, 4).EQ.0) GO TO 10
DELAY = WEEK + 52 - DUE
GO TO 20
10 DELAY = WEEK - DUE
Determine Equipment Performance
20 IDelay = DELAY
IF(Delay.LE.20)GO TO 30
DELAY = 20
30 IF(Delay.GE.9)GO TO 40
IDELAY = -9
40 DELAY = DELAY
EQUPERF(PROD, DELAY) = EQUPERF(PROD, DELAY) + QTY
IF(ISPRINT.GT.1)GO TO 45
WRITE(2, 43) PROD, DELAY, ORDNO, ITEM, OPFILE(KEYNO, ITEM, M, 3), QTY
43 FORMAT(1HO, 6X, 'EQUIPMENT DELIVERY PERFORMANCE FOR PRODUCT NO., 16, 2X, DELAY = ', 16, 2X)
45 IF(OPFILE(KEYNO, ITEM, M, 2).NE.OPFILE(KEYNO, ITEM, M, 9)) GO TO 100
ITEMPERF(PROD, DELAY) = ITEMPERF(PROD, DELAY) + 1
IF(ISPRINT.GT.1)GO TO 49
WRITE(2, 47) PROD, DELAY
47 FORMAT(1HO, 6X, 'ITEM DELIVERY PERFORMANCE FOR PRODUCT NO., 16, 2X, DELAY = ', 16)
49 IF(OPFILE(KEYNO, ITEM, M, 2).NE.OPFILE(KEYNO, ITEM, M, 9)) GO TO 100
IF(OPFILE(KEYNO, ITEM, M, 2).NE.OPFILE(KEYNO, ITEM, M, 9)) GO TO 100
50 CONTINUE
ORDPERF(1, DELAY) = ORDPERF(1, DELAY) + 1
IF(ISPRINT.GT.1)GO TO 55
WRITE(2, 55) DELAY
55 FORMAT(1HO, 6X, 'ORDER PERFORMANCE (PART SHIP) DELAY = ', 16)
2 15, 2X, 'QTY', 15, 2X, 'DESPATCHED'
CHECK FOR PART SHIPMENT
49 IF(ORDBK(PROD, DUE, N, 5).EQ.2) GO TO 60
Determine Order Performance = Part Ship
I DUE = OPFILE(KEYNO, ITEM, M, 3)
DO 50 J = 1, 5
DO 50 J = 1, 5
IF(OPFILE(KEYNO, ITEM, M, 3).NE.IDUE)GO TO 50
IF(OPFILE(KEYNO, ITEM, M, 2).NE.OPFILE(KEYNO, ITEM, M, 9)) GO TO 100
50 CONTINUE
ORDPERF(1, DELAY) = ORDPERF(1, DELAY) + 1
IF(ISPRINT.GT.1)GO TO 56
WRITE(2, 56) DELAY
56 FORMAT(1HO, 6X, 'SHIP GOODS')
WRITE(2,200) OPFILE(KEYNO,ITEM,M,10), IDUE
200 FORMAT(1HO,6X, 1ORDER NO., 16, 2X, 'DUE WEEK', 16, 2X, 'DESPATCHED')
56 DO 57 I=1,5
57 J=1,5
IF(OPFILE(KEYNO,I,J,3).NE.IDUE) GO TO 57
OPFILE(KEYNO,I,J,9)=0
57 CONTINUE

C CHECK IF ORDER IS COMPLETE
C
IOTY=0
DO 59 I=1,5
DO 59 J=1,5
DO 58 K=0,9
58 IOTY=IOTY+OPFILE(KEYNO,I,J,K)
IF(IOTY.NE.0) GO TO 100
59 CONTINUE
GO TO 90

C DETERMINE ORDER PERFORMANCE - NO PART SHIP
C
60 IMAX=0
DO 70 I=1,5
DO 70 J=1,5
IF(OPFILE(KEYNO,I,J,2).NE.OPFILE(KEYNO,I,J,9)) GO TO 100
IF(OPFILE(KEYNO,I,J,3).LE.IMAX) GO TO 70
IMAX=OPFILE(KEYNO,I,J,3)
70 CONTINUE
IDELAY=DELAY+OPFILE(KEYNO,ITEM,M,3)-IMAX
IF(IDELAY.LE.20) GO TO 73
IDELAY=20
73 IF(IDELAY.GE.9) GO TO 77
IDELAY=9
77 IDELAY=21-IDELAY
ORDPERF(2,IDELAY)=ORDPERF(2,IDELAY)+1
IF(IPRINT.GT.2) GO TO 90
WRITE(2,80) DELAY
80 FORMAT(1HO,6X, 1ORDER PERFORMANCE (NO PART SHIP), DELAY =1,16)
WRITE(2,300) OPFILE(KEYNO,ITEM,M,10)
300 FORMAT(1HO,6X, 1ORDER NO., 16, 2X, 'DESPATCHED')

C CLEAR DOWN OPFILE AND ORDBK
C
90 DO 110 I=1,5
90 J=1,5
90 K=1,10
110 OPFILE(KEYNO,I,J,K)=0
DO 120 IPROD=1,5
105 DO 120 L=1,52
120 CONTINUE
100 RETURN
END
B6 - SUBROUTINE DESPATCH

Equipments which have been tested are moved to the warehouse, which is regarded as commercial stock, by subroutine DESPATCH. Two alternative priority rules which determine the sequence of processing are available; first-in-first-out or by due date.

Equipments for despatch are contained in the file TESTQ. Selection of the due date priority rule will cause TESTQ to be sorted into due date sequence prior to processing. Bypassing the sort segment will result in first-in-first-out logic.

The size of the delivery batch is determined by the delivery plan contained in file DELPLAN for the appropriate period, assuming that sufficient equipments are eligible for delivery. If the delivery capacity is under-utilised due to insufficient eligible equipments in the queue, the excess capacity is considered lost and cannot be carried forward into future periods.

Items are selected sequentially from TESTQ, and the appropriate OPFILE and ORDBK records located. Equipments are not eligible for delivery within a minimum period of three days of entering the test queue, recognising a realistic expedited throughput time. Each item selected will have the whole or part of the batch quantity despatched, this quantity being dependent upon the balance of the delivery plan available.

Successful despatch of a quantity will cause a number of files to be updated, reflecting the changed status of the order.

a) TESTQ will have the quantity balance reduced by the quantity despatched.

b) OPFILE will show the despatched quantity as having moved from TEST to DESPATCH status.

c) ORDBK will show a reduction in the outstanding balance by the quantity despatched.
d) TOTAL will show a reduction in total load for non-overdue orders.

e) The balance of the delivery plan, DELPLAN, for the appropriate week will be reduced by the despatched quantity.

If a finite delivery plan balance remains the next item will be selected from TESTQ. If TESTQ has no further eligible orders, or the delivery plan has been exhausted, TESTQ will be reset by removing fully despatched items and consolidating the balance of the file.
SUBROUTINE DESPATCH

PRIORITY RULE?

FIFO

SELECT NEXT PRODUCT

ESTABLISH DELIVERY QTY FROM DELPLAN

QTY ≤ 0

SELECT NEXT ITEM IN TEST QUEUE

ESTABLISH ITEM QTY

N

ITEM QTY > DELPLAN?

Y

QUANTITY = ITEM QTY

SUBROUTINE DELPERF

REMOVE QUANTITY FROM TEST QUEUE

UPDATE ORDBK/OPFILE STATUS

ESTABLISH DELIVERY PERFORMANCE

REDUCE DELPLAN BY QUANTITY

DELPLAN = 0

N

PRINT "EXHAUSTED" ORDERS IN TEST

REORGANISE TEST QUEUE

RETURN

Summarised Flow Chart - Subroutine Despatch
SUBROUTINE DESPATCH(TESTQ, OPFILE, WEEK, DELPLAN, IFILE, ORDBK, EQUPERF;
ITEMPERF, IORDPERF, TOTAL, IPRINT, AFILE, REFWEEK, ISORT, DIAGFILE, RUNTIME
2E)
REAL AFILE(25, 3)
INTEGER TESTQ(5, 50, 50), OPFILE(125, 5, 50, 10), WEEK, DELPLAN(5, 50, 50), IFILE
(25, 25), ORDBK(5, 50, 20, 5), EQUPERF(10, 30), ITEMPEAF(10, 30), IORDPERF(2
25, 30), TOTAL(5, 52), REFWEEK, QTY, DIAGFILE(10, 2), RUNTIME
IF(IPRINT.GT.1) GO TO 1
WRITE(2, 2)
2 FORMAT(1HO, 6X, ' * * * * DESPATCH * * * *')
SORT TEST QUEUE
1 IF(ISORT.EQ.0) GO TO 4
DO 3 I=1, 5
DO 3 J=1, 49
DO 3 K=1, 50
IF(TESTQ(I, J, K, 4).EQ.0) GO TO 3
IF(TESTQ(I, J, K, 3).LE.TESTQ(I, K, 3)) GO TO 3
DO 3 L=1, 5
M=TESTQ(I, J, L)
TESTQ(I, J, L).TESTQ(I, K, L)
TESTQ(I, K, L) = M
3 CONTINUE
4 DO 200 I=1, 5
IF(IFILE(I, 11).NE.1) GO TO 200
IF(DELP(AN(I, WEEK).LE.0) GO TO 200
IF(IPRINT.GT.2) GO TO 5
WRITE(2, 250) AFILE(I, 1), DELPLAN(I, WEEK)
520 FORMAT(1HO, 6X, ' DELIVERY BATCH FOR ', A8, ', I = 1; 15)
5 DO 60 J=1, 50
IF(TESTQ(I, J, 5).EQ.0) GO TO 60
IF(TESTQ(I, J, 4).EQ.0) GO TO 60
10 JA=TESTQ(I, J, 1)
JB=TESTQ(I, J, 2)
JC=TESTQ(I, J, 3)
DO 15 KEYNO=1, 125
IF(OPFILE(KEYNO, JB, 1, 10).EQ. JA) GO TO 17
15 CONTINUE
GO TO 25
17 DO 20 M=1, 5
IF(OPFILE(KEYNO, JB, M, 3).EQ. JC) GO TO 31
20 CONTINUE
25 IF(IPRINT.GT.5) GO TO 60
WRITE(2, 30) JA, JB, JC, OPFILE(KEYNO, JB, M, 3), REFWEEK
30 FORMAT(1HO, 6X, 'INCOMPATIBLE DATA IN OPFILE FOR ORDER NO. ', I4, ', ITE
1M NO. ', I4, ', DUE WEEK ', I4/6X, 'ACTUAL WEEK ', I4, 'REFWEEK ', I4)
IPRINT=1
DO 29 IZ=1, 10
29 DIAGFILE(IZ, 2)=1
RUNTIME=REFWEEK+WEEK+4
GO TO 60
31 NOD=0
IF(JC.GT.REFWEEK) GO TO 32
NOD=52
32 JD=JC-REFWEEK+NOD
DO 33 N=1, 20
IF(ORDBK(I, JD, N, 1).NE. JA) GO TO 33
IF(ORDBK(I, JD, N, 2).EQ. JB) GO TO 40
33 CONTINUE
IF(IPRINT.GT.5)GO TO 60
WRITE(2,35)JA,JND,AFILE(I,1),7C
35 FORMAT(1HO.6X,'INCOMPATIBLE DATA IN ORDER BOOK FOR ORDER NUMBER',I16.2X,'DUE WEEK',I16.2X,'FOR PRODUCT NUMBER',A8,2X,'ACTUAL WEEK',I16)
IPRINT=1
DO 37 IZ=1,10
37 DIAGFILE(IZ,2)=1
RUNTIME=REFWEEK+WEKP+4
GO TO 60
40 IF(TESTQ(I,J,4).LT.DELPLAN(I,WEEK)).GO TO 45
QTY= DELPLAN(I,WEEK)
GO TO 50
45 QTY = TESTQ(I,J,4)
50 TESTQ(I,J,4)=TESTQ(I,J,4)-QTY
IF(FILES(I,J,19))=IFILES(I,J,19)-QTY
OPFILE(KEYNO,JBD,M,9)=OPFILE(KEYNO,JBD,M,9)+QTY
OPFILE(KEYNO,JBD,M,8)=OPFILE(KEYNO,JBD,M,8)-QTY
ORDBK(I,J,D,N,3)=ORDBK(I,J,D,N,3)-QTY
IF(ORDBK(I,J,D,N,4).EQ.1)GO TO 55
55 TOTAL(I,D)=TOTAL(I,D)-QTY
50 CALL DELPERF(ORDBK,OPFILE,I,KEYNO,JBD,J,D,WEKP,N,N,IPRINT,EQU PERF,IT
1 TEMPERF,ORDPERF,QTY)
DELPLAN(I,WEEK)=DELPLAN(I,WEEK)-QTY
IF(DELPLAN(I,WEEK).EQ.0).GO TO 150
60 CONTINUE
500 IF(IPRINT.GT.2).GO TO 150
WRITE(2,510)AFILE(I,1)
510 FORMAT(1HO.6X,'EXHAUSTED ORDERS IN TEST FOR PRODUCT',A8,2X)
150 DO 160 MM=1,49
IF(TESTQ(I,MM,4).GT.0).GO TO 160
153 CONTINUE
160 CONTINUE
155 CONTINUE
158 CONTINUE
160 CONTINUE
200 CONTINUE
IF(IPRINT.GT.3).GO TO 250
CALL TOPRINT(TESTQ,WEEK,AFILE,REFWEEK)
250 RETURN
END
Sub-assembly programmes are based on forecast quarterly demand and use a simple "Economic Batch Quantity" formula to determine the size and frequency of batches within the quarter.

The total quarterly demand for the sub-assembly is derived from the file REQFILE, following which the number of batches in the quarter is calculated. The number of batches per annum is derived from the standard EBQ formula.

\[
N = \frac{\sqrt{\frac{C_l}{2C_3}}}{r}
\]

where \( r = \text{demand/annum} \)

\( C_l = \text{holding cost/unit} \)

\( C_3 = \text{set up cost} \)

If \( q \) is the quarterly demand and \( C' \) is the unit value, it is assumed that the interest rate is 10% and the set up cost is £10.*

Thus:

\[
N = \sqrt{\frac{q x 0.1 x C'}{2 x 10}}
\]

\[
= \sqrt{0.02qC'}
\]

The unit value is derived from IFILE by summing the material and labour value. The number of batches per quarter is thus

\[
n = \frac{1}{4} \sqrt{0.02qC'}
\]

Since it is required to phase the batches evenly throughout the thirteen week period, only integer values of 1, 2, 3, 4, 6 or 13 are permissible as the number of batches (\( n' \)).

The nominal "economic batch quantity" is the quarterly demand divided by \( n' \). The cycle time between batches is \( 13/n' \). Both parameters are retained in IFILE for subsequent use in the sub-assembly programme SUBPROG.

* Note: Values currently used in practice.
SUBROUTINE EBQ

SELECT NEXT PRODUCT

SUB-ASSEMBLY ?

Y

CALCULATE DEMAND FOR QUARTER AND INCREMENT CAPACITY

CALCULATE EBQ AND CYCLE TIME

ENTER PARAMETERS IN ITEM MASTER

EVALUATE GROSS AND NETT CAPACITY

RETURN

Summarised Flow Chart - Subroutine EBQ
SUBROUTINE EDQCRE
(IFILE, IPRINT, AFILE, CAPACITY, UTIL)
REAL AFILE(25,3)
INTEGER IFILE(25,23), REQFILE(25,52), WEEK, DEMAND, CAPACITY
IC=0
DO 30 I=1,25
IF (IFILE(I,11), EQ, 2) GO TO 5
GO TO 30
5 DEMAND=0
DO 10 J=1,13
IF (IFILE(I,15), EQ, 1) GO TO 10
IC=IC+REQFILE(I,J)
10 DEMAND=DEMAND+REQFILE(I,J)
N=NINT(SQRT(DEMAND*IFILE(I,13)*IFILE(I,14)*0.02)/4.0)
IF (N.GT.0) GO TO 15
N=1
15 IF (N.LE.4) GO TO 20
IF (N.GT.6) GO TO 17
N=6
GO TO 20
17 N=13
20 IFILE(I,21)=DEMAND/N
IF (FILE(I,22)=NINT(13.0/N)
IF (IPRINT.GT.4) GO TO 30
WRITE(2,50)AFILE(I,1), IFILE(I,21), IFILE(I,22)
50 FORMAT(1HO,6X, 'SUB-ASSY NO. ', A8, 6X, 'EDQ', 15, 6X, 'CYCLE TIME', I5)
30 CONTINUE
CAPACITY=IFIX(IC/UTIL)
IF (IPRINT.GT.3) GO TO 70
WRITE(2,60)IC, UTIL, CAPACITY
60 FORMAT(1HO,6X, IP, C.D; CAPACITY ..., CAPACITY REQD = 1,16/27X, 1'UTILISATION = 1,4.2/27X', 'GROSS CAPACITY = 1,16)
70 RETURN
END
Under normal conditions, the assumption has been made that material will arrive from the supplier according to the call-off schedule modified by a probabilistic element within the subroutine RECEIVE. Within the logic of the model, the assumption is maintained provided there is a finite level of free stock to service production requirements. If, however, the level of free stock falls below zero, the item concerned is considered to change from the normal "unexpedited" mode to "expedite" mode.

The prime objective of a material expeditor, or "progress chaser", is to obtain shortage items at the earliest possible opportunity. This success will depend, for example, upon whether or not the item can be procured through secondary sources, the current load on the supplier, the notice given to the supplier to respond and any other items that could possibly be "unexpedited" to release capacity for the urgent item.

In practice there are three levels of expediting:

a) the item is a "line hold", which signifies that a batch of material with a known (or subsequently discovered) shortage is in work in progress and is a probable line stopper.

b) the item is a "shortage to allocation", which implies that free stock has become negative.

c) the item is in arrears to the supplier call-off schedule.

All three conditions may exist concurrently.

The line hold situation is considered in the model to be covered by the shortage to allocation activity. Any item which becomes a line hold but was not identified as a shortage to allocation previously would normally be as a result of incorrect material being issued, a stock record error, a bill of material error, a late engineering change or a scrap problem.
None of these eventualities has specifically been addressed by the model.

The arrears to schedule situation has been considered in the establishment of an input profile within subroutine INPUT.

Subroutine EXPEDITE, therefore, considers only the shortage to allocation, or negative free stock situation.

The stock file record within the item master file, IFILE, is scanned at the start of a weekly cycle. The existence of negative free stock for a bought-out (level 4) part, will cause the expediting sequences to be invoked.

The quantity short, which is calculated as allocated stock less physical stock, is compared with the forward input schedule.

When an input batch has been identified, file EXFILE is interrogated to ascertain whether the batch has been subject to previous expediting activity. The assumption is made that a batch may be expedited once only.

If the batch is not already subject to expediting action, the next step is to determine the amount of time that the batch can be pulled forward. Factual information upon which to base an algorithm is highly complex, depending on such factors as the component type (proprietary item "ex stock" or made to order), the imminence of the next schedule receipt, the existence of a schedule arrear, the existence of a supplier order, the expeditor/supplier relationship and many other considerations.

The objective of the model was to arrive at a reasonable and quantifiable algorithm which could be simply applied. The first stage is to sample from the histogram defined in the table below.

<table>
<thead>
<tr>
<th>Value</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

Expedite Histogram
The value generated is the "degree of success" of the expediting effort.

The time remaining before the next batch is due to arrive is divided by the above factor to determine the number of weeks pulled forward. For example, if the current week is 4 and the next batch is due week 8, the time remaining is 4 weeks. If the value sampled is 2, the number of weeks pulled forward is 4/2, i.e. 2 weeks.

The input schedule is then modified to reflect the pull forward action and file EXFILE is correspondingly tagged to inhibit any further activity on the expedite batch.

If the first batch is insufficient to satisfy the shortage, further batches are sought and actioned following the same logic.

If no input batch can be identified for the next 13 weeks, the assumption is made that any item can be procured on an eight week lead time. This is a prime assumption, since no account is taken of the type of commodity and the degree of urgency, but is not unrealistic as an average (assuming that the item is not relating to a new product.)

Where expediting action is required and a batch (or batches) has already been subject to a previous expedite, the value in EXFILE for the appropriate week is incremented by one. Thus, the modeller may access the number of times that a part is expedited as part of the analysis of simulation results.
Summarised Flow Chart - Sub-routine Expedite
SUBROUTINE EXPEDITE(IFILE,WEEK,INPUT,ISEED3,TABLE3,IPRINT,EXFILE,AFILE)
REAL AFILE(25,3)
INTEGER IFILE(25,23),INPUT(25,52),SHORT,WEEK TABLE3(7,2),EXFILE(25,1752)
IF(ISEED3.EQ.0)GO TO 250
IF(IPRINT.GT.1)GO TO 10
WRITE(2,20)
20 FORMAT(IHO,6X,1** EXPEDITE * * * *//)
10 DO 100 I=1,25
IF(IFILE(I,1).NE.4)GO TO 100
SHORT=IFILE(I,17)+IFILE(I,16)
IF(SHORT.LE.0)GO TO 100
C FIND NEXT BATCH INPUT
DO 50 J=WEEK,WEEK+13
IF(INPUT(I,J).EQ.0)GO TO 50
IF(EXFILE(I,J).EQ.0)GO TO 30
EXFILE(I,J)=EXFILE(I,J)+1
IVAL=INPUT(I,J)
IF(IPRINT.GT.2)GO TO 32
WRITE(2,150)AFILE(I,J),SHORT,INPUT(I,J)
150 FORMAT(IHO,6X,A8,6X,DUE WEEK',I5,6X,ALREADY EXPEDITED!)
GO TO 40
30 CALL HSAMPLE(I,ISEED3,NUMBER,TABLE3)
K=J+NINT(FLOAT(J-WEEK)/NUMBER)
IF(IPRINT.GT.2)GO TO 32
WRITE(2,150)AFILE(I,J),SHORT,INPUT(I,J),NUMBER,K
150 FORMAT(IHO,6X,A8,6X,J=I,14,3X,SHORT=I,14,3X,INPUT(I,J)=I,14,3X,NUMBER=I,14,3X,K=I,14)
32 IF(K.GT.0)GO TO 35
K=1
35 IVAL=INPUT(I,J)
INPUT(I,K)=INPUT(I,K)+IVAL
EXFILE(I,K)=EXFILE(I,K)+1
SHORT=SHORT-IVAL
40 IF(SHORT.LE.0)GO TO 100
50 CONTINUE
100 CONTINUE
250 RETURN
END
The parts explosion, or Material Requirements Planning logic, is contained within the subroutine EXPLODE.

The subroutine has access to the basic relationship and stock status data held within the item master file, IFILE. The prime input data is the requirements file, REQFILE, for each part.

To ensure that all requirements have been registered for a part before subsequent explosion takes place, a "low level code" is held in the item master file indicating the lowest level that the part is to be found in any product. Level 1 denotes the top level product, levels 2 and 3 are intermediate sub-assemblies and level 4 denotes purchased parts.

The requirements planning process takes place on a level by level basis, starting at level 1.

For each item, the free stock is calculated and a net plan derived from the subroutine NETTING. Since the requirements data for a top level product (level 1) is the off-line plan and does not include finished equipment stock, the available stock is taken to be work-in-progress only.

Following the netting process, the resultant plan is off-set by the lead time for "in house" manufactured items. Any "overspill" is held in the first period of the plan.

Each component part of the assembly is then found by reference to the item master record and the requirements file for the component is increased by the generated requirement extended by the quantity per.

The above logic is a reasonable approximation to that incorporated into the existing computer system, with the exception that any netting of intermediate sub-assemblies must be manually input to the system as net + or - value.
An example of the requirements planning logic is shown below.

![Diagram showing requirements planning logic]

<table>
<thead>
<tr>
<th>PART &quot;A&quot;</th>
<th>PERIOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>REQUIREMENTS</td>
<td>10</td>
</tr>
<tr>
<td>NET PLAN</td>
<td>-</td>
</tr>
<tr>
<td>LEAD TIME OFFSET</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PART &quot;B&quot;</th>
<th>PERIOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>REQUIREMENTS</td>
<td>-</td>
</tr>
<tr>
<td>NET PLAN</td>
<td>-</td>
</tr>
<tr>
<td>LEAD TIME OFFSET</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PART &quot;C&quot;</th>
<th>PERIOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>REQUIREMENTS</td>
<td>-</td>
</tr>
<tr>
<td>NET PLAN</td>
<td>-</td>
</tr>
<tr>
<td>LEAD TIME OFFSET</td>
<td>-</td>
</tr>
</tbody>
</table>
SUBROUTINE EXPLODE

LOW LEVEL CODE = 1

SELECT NEXT PRODUCT

ESTABLISH QTY IN ALLOCATION

CALCULATE AVAILABLE STOCK

ESTABLISH NETT REQUIREMENTS

COMPONENT LEVEL?

Y

OFFSET FOR LEADTIME

ADD SCHEDULE TO REQUIREMENTS OF COMPONENT ITEMS

ESTABLISH SCHEDULE FOR THIS ITEM

LEVEL = LEVEL + 1

LEVEL <= LEVEL

V

N

RETURN

Summarised Flow Chart - Subroutine Explode
SUBROUTINE EXPLODE( AFILE, IFILE, REQFILE, IPRINT, ALLQ)
REAL AFILE(25,3)
INTEGER LEVEL, IFILE(25,23), BUFFER, AVAIL, REQFILE(25,52), QTY, WORKFILE(1E(52), ALLO(12,50,6), ALLQTY
LEVEL=1
10 DO 100 I=1,25
   IF( FILE(I,11).NE.;LEVEL) GO TO 100
   IF(LEVEL.EQ.4) GO TO 30
   ALLQTY=0
   DO 20 J=1,50
20   ALLQTY=ALLQTY+ALLQ(I,J)
   DO 40 J=1,52
40   WORKFILE(J)=REQFILE(I,J)
   IF( IPRINT.GT.3) GO TO 50
   WRITE(2,510) AFILE(I,1)
500   FORMAT(1HO.12X,'REQUIREMENTS FOR PART NUMBER',2X,A8,'.*.*.*')
   WRITE(2,510)( WORKFILE(J), J=1,51)
510   FORMAT(1HO.1~X,'REQUIREMENTS',16X,20I15.2X,2I5.2X,'AVAILABLE =',15.2X).
   AVAIL=IFILE(I,18)
   BUFFER=AVAIL
   GO TO 57
   IF(LEVEL.EQ.4) GO TO 95
   M=52-IFILE(I,12)+2
   DO 60 K=1, M
60   WORKFILE(K+1)=WORKFILE(K+1)+WORKFILE(K)
   DO 70 K=1,M
70   WORKFILE(K)=WORKFILE(K+1)+WORKFILE(I,12)+2
   IF( IPRINT.GT.2) GO TO 75
   WRITE(2,530)( WORKFILE(J), J=1,52)
530   FORMAT(1HO.12X;'OFFSET FOR LEAD TIME ::'15.2X,'PLUS ALLOCATION OF TWO WEEKS',15.2X,'NET PLAN',6X,20I5.2X)
   IF(LEVEL.EQ.4) GO TO 95
   L=1
   DO 90 N=FILE(I,L)
90   IF(N.EQ.0) GO TO 91
   QTY=FILE(I,L+1)
   DO 90 M=1,52
90   REQFILE(N,M)=REQFILE(N,M)+WORKFILE(M)*QTY
   L=L+2
   IF(L.GT.9) GO TO 100
   GO TO 80
   CONTINUE
   LEVEL=LEVEL+1
   IF(LEVEL.LE.4) GO TO 10
100 RETURN
END
Customer orders are selected, in subroutine ALLOCATE, to be taken from stock or to be made from components. Items to be taken from stock are moved to "ex-stock" status in OPFILE and the free stock of the appropriate product is reduced.

Subroutine FESALL will subsequently "allocate" a stock equipment to the customer order and move the order to "test" status in OPFILE. The level of finished equipment stock will also be reduced as stock is moved.

The logic within FESALL is to select each product in turn, and to follow the identical processing sequence twice; once for overdue orders and again for due orders.

For overdue orders, the search horizon is set to 52 weeks and the processing index set to "1". ORDBK is searched for any outstanding orders with the flag set to overdue. If an order is found, the true due date is calculated and the corresponding record selected from OPFILE. If the record cannot be located in OPFILE, an error condition is reported, the diagnostic level set to full detail and the simulation run terminated after a further few weeks. Successful location of the correct record in OPFILE is then followed by a check on the order status. If an order is found in "ex-stock" status, the following activities are initiated, otherwise the search continues to the end of the search horizon.

For orders in "ex-stock" status, the quantity required is compared with the physical stock for the product as held in IFILE. The quantity is set to the lesser of the requirements and the physical stock;

- the quantity is moved from "ex-stock" to "test" status in OPFILE.
- physical stock and allocated stock are reduced by the quantity moved.
- test work in progress is augmented by the quantity moved.
- a record is added to TESTQ, defining the order number, item, due date and quantity.
- the TESTQ timer is set to the minimum throughput time of 3 days.

The above process is repeated until either stock or available orders within the search horizon are exhausted.

The process is repeated for due orders, where the index is set to "0" and the search horizon is set to the current week plus two.

Summarised Flow Chart - Subroutine Fesall

* SEGMENT REPEATED
  1) OVERTIME ITEMS
  2) DUE ITEMS
SUBROUTINE FESALL(ORDBK, OPFILE, WEEK, IFILE, AFILE, IPRINT, REFWEEK, TESTQ, DIAGFILE, RUNTIME)
REAL AFILE(25,3)
INTEGER ORDNO, IFILE(25), TESTQ(550,5), REFWEEK, WEEK, DIAGFILE(10,2), RUNTIME
IF( IPRINT.GT.1) GO TO 300
WRITE(2,320)
320 FORMAT(1HO,6X,'*** FESALL *** */')
300 DO 200 I=1,5
JJ=52
INDEX=1
100 DO 90 J=1,5
DO 80 K=1,20
If(ORDBK(I,J,K;3):EQ;0) GO TO 60
IF(ORDBK(I,J,K;4);NE;INDEX) GO TO 60
DUEDATE=J
IF(INDEX.EQ.0) GO TO 2
DUEDATE=J-52
2 ORDNO=ORDBK(I,J,K;1)
ITEMNO=ORDBK(I,J,K,2)
DO 1 KEYNO=1,125
IF(OPFILE(KEYNO,ITEMNO;1,10),EQ,ORDNO) GO TO 3
1 CONTINUE
GO TO 7
3 DO 5 L=1,5
4 IF((OPFILE(KEYNO,ITEMNO;L,3)-REFWEEK),EQ,DUEDATE) GO TO 20
5 CONTINUE
7 IF(IPRINT.GT.5) GO TO 60
WRITE(2,40) I
40 FORMAT(1HO,6X,'INCOMPATIBLE DATA IN ORDBK AND OPFILE FOR ORDER ',I6,' ITEM ',I6,' WEEK ',16,' ACTUAL WEEK ',14,' REFWEEK ',1,14)
PRINT=1
DO 15 !=1,10
15 DIAGFILE(IZ,2)=1
RUNTIME=REFWEEK+WEK+4
GO TO 60
20 IF(OPFILE(KEYNO,ITEMNO;L;5),EQ;0) GO TO 60
DO 30 M=1,50
IF(TESTQ(I,M;1),EQ;0) GO TO 45
30 CONTINUE
IF(IPRINT.GT.5) GO TO 200
WRITE(2;40) I
40 FORMAT(1HO,6X,'EXCEEDED TEST QUEUE SIZE FOR PRODUCT ',I6)
GO TO 200
45 QTY=OPFILE(KEYNO,ITEMNO;L,5)
IF(OPFILE(KEYNO,ITEMNO;L,5),LT,IFILE(I,16)) GO TO 50
QTY=IFILE(I,16)
50 OPFILE(KEYNO,ITEMNO;L,5)=OPFILE(KEYNO,ITEMNO;L,5)+QTY
OPFILE(KEYNO,ITEMNO;L,8)=OPFILE(KEYNO,ITEMNO;L,8)+QTY
TESTQ(I,M;1)=ORDNO
TESTQ(I,M;2)=ITEMNO
TESTQ(I,M;3)=OPFILE(KEYNO,ITEMNO;L,3)
TESTQ(I,M;4)=QTY
TESTQ(I,M;5)=3
IFILE(I,19)=IFILE(I,19)-QTY
IFILE(I,16)=IFILE(I,16)-QTY
IF(IPRINT.GT.2) GO TO 55
WRITE(2,350) ORDNO, ITEMNO, OPFILE(KEYNO,ITEMNO;L,3), AFILE(I,1), QTY
350 FORMAT(1HO,6X,'ORDER ',I6,' ITEM ',I6,' DUE ',I6,' PROD ',A,4X)
17,' QTY ',I6,' ALLOCATED AGAINST STOCK')
55 IF(FILE(I,16).LE.0)GO TO 200
60 CONTINUE
   IF(INDEX.EQ.0)GO TO 200
   JJ=WEEK+2
   INDEX=0
   GO TO 100
200 CONTINUE
   RETURN
   END
B11 - SUBROUTINE FORECAST

To support the preparation of the quarterly plan QPLAN, a mechanism for forecasting the orders to be received in future periods is required.

The present method of orders received forecasting is to use as the base a 12 month moving annual total modified by any specific knowledge relating to markets, competition, large order opportunities and other considerations.

The model requires a simple form of forecasting mechanism which may respond to moderate levels of variability and trend as introduced by the orders generator. This mechanism is provided by a basic linear regression method.

Linear regression will determine the "line of best fit" through a number of points, where the "best fit" is that line which minimises the square of the errors. The error is the distance between each observation and the regression line.

If the equation for a straight line is given by

\[ x = a + bt \]

where \( x \) = demand
\( t \) = time

the square of the errors is minimised when

\[ a = \frac{(\sum x \cdot \sum t^2) - (\sum t \cdot \sum x \cdot t)}{(n \cdot \sum t^2) - (\sum t)^2} \] \hspace{1cm} (i)

\[ b = \frac{n \cdot \sum x \cdot t - (\sum x \cdot \sum t)}{(n \cdot \sum t^2) - (\sum t)^2} \] \hspace{1cm} (ii)

or \[ a = \frac{\sum x - b \sum t}{n} \] \hspace{1cm} (iii)
The model calculates a regression line based on 12 monthly observations derived from the order history file ORDHIST. The observations contained within ORDHIST are the demand data, considered in the model as "Y" axis observations. The "X" axis observations correspond to time and represent periods 1 - 12 inclusive.

Following initialisation, the program calculates the sums

\[
\begin{align*}
\text{SUMX} & = \sum_{i=1}^{12} x_i \\
\text{SUMX}_2 & = \sum_{i=1}^{12} x_i^2 \\
\text{SUMY} & = \sum_{i=1}^{12} y_i \\
\text{SUMY}_2 & = \sum_{i=1}^{12} y_i^2 \\
\text{SUMXY} & = \sum_{i=1}^{12} i \cdot x_i
\end{align*}
\]

where \(x_i\) is the \(i^{th}\) observation from ORDHIST.

These are then substituted in equations (ii) and (iii) to give

\[
\begin{align*}
B & = \frac{12 \cdot \text{SUMXY} - \text{SUMX} \cdot \text{SUMY}}{12 \cdot \text{SUMX}_2 - \text{SUMX}^2} \\
A & = \frac{\text{SUMY} - B \cdot \text{SUMX}}{12}
\end{align*}
\]
The equation of the regression line is then given by

\[ Y = BX + A \]

This straight line is then extrapolated to provide a forecast of orders to be received in future periods.

The diagram above shows that the first period to be forecast is for ORF(2) which contains months 13 through 15. Thus,

\[ \sum_{i=13}^{15} x_i = 15 \]

where \( x \) is the monthly observation given by

\[ Y = BX + A \]

and \( X \) is the month number.

Therefore,

\[ \sum_{j=13}^{15} B_j + A = j = 15 \]
Similarly,

\[ \text{ORF}(3) = \sum_{j=16}^{18} B \cdot j + A \]

etc.
FORECAST

SUBROUTINE FORECAST(ORDHIST, ORF, IPRINT, AFILE, IFILE)
REAL AFILE(25,3)
INTEGER ORF(10,7), IORDHIST(5,12), IFILE(25,23)
IF(IPRINT.GT.1) GO TO 5
WRITE(2,200)
200 FORMAT(1HO, 6X, '* ***' * FORECAST *** */)

CALCULATE REGRESSION LINE

5 DO 150 M = 1, 5
   IF(IPRINT(M,11).NE.1) GO TO 150
   SUMX=0.0
   SUMX2=0.0
   SUMXY=0.0
   SUMY=0.0
   SUMY2=0.0
   DO 10 I=1,12
      SUMX=SUMX+FLOAT(I)
      SUMX2=SUMX2+FLOAT(I)**2
      SUMY=SUMY+FLOAT(ORDHIST(M,I))
      SUMY2=SUMY2+FLOAT(ORDHIST(M,I)**2)
   10 SUMXY=SUMXY+FLOAT(ORDHIST(M,I))*I
   B=(12*SUMXY-SUMX*SUMY)/(12*SUMX2-SUMX**2)
   A=(SUMY-B*SUMX)/12
   C=SUMY/12
   IF(IPRINT.GT.2) GO TO 70
   WRITE(2,20) AFILE(1,1)
   20 FORMAT(1HO, 6X, 'EQUATION OF REGRESSION LINE FOR PRODUCT ', A, ' IS')
   IF(A) 30, 50
   30 WRITE(2,40) A
   40 FORMAT(1HO, 6X, 'Y = ', F6.1, 'X + F6.1')
   50 WRITE(2,60) A
   60 FORMAT(1HO, 6X, 'Y = ', F6.1, 'X + ', F6.1)

CALCULATE FORECAST

70 DO 120 I=2,7
   ORF(M,1)=0
   J=3*I+7
   K=J+2
   DO 110 L=J,K
      ORF(M,L)=ORF(M,L)*IFIX(B*L+A)
   110 IF(ORF(M,L).GE.0) GO TO 120
   ORF(M,1)=0
120 CONTINUE
   IF(IPRINT.GT.3) GO TO 150
   WRITE(2,130) (ORDHIST(M,I), I=1,12), (ORF(M,I), I=2,7)
130 FORMAT(1HO, 6X, 'HISTORY (MONTHLY)!'//6X,1216//6X, 'FORECAST (QUARTERLY)'!//6X,616///)
150 CONTINUE
RETURN
END
Subroutine HSAMPLE is used to generate random variates as defined by a predetermined histogram. Two subroutines are available; HSAMPLE 1 which samples from a 2 x 7 array, and HSAMPLE 2 which samples from a 2 x 2 array.

The arrays containing the histogram parameters are contained in cumulative form.

The subroutine derives the array (or table) and the random number generators seed from the calling segment. A uniformly distributed random variate defined on the unit interval is expanded to the scale 0 - 100 and the corresponding histogram interval is sought. The value associated with the histogram interval is then returned to the calling segment.

Thus, a sample from RANDOM of, say, 0.84 would provide, from the histogram, a value of 5.
SUBROUTINE HSAMPLE 1(J, VALUE; TABLE)
INTEGER TABLE(7, 2), VALUE
CALL RANDOM(E, J)
NUMBER = 100*E
K = 0
10 K = K + 1
   IF (NUMBER .GT. TABLE(K, 2)) GO TO 10
   VALUE = TABLE(K, 1)
RETURN
END

SUBROUTINE HSAMPLE 2(J, VALUE; TABLE)
INTEGER TABLE(2, 2), VALUE
CALL RANDOM(E, J)
NUMBER = 100*E
K = 0
10 K = K + 1
   IF (NUMBER .GT. TABLE(K, 2)) GO TO 10
   VALUE = TABLE(K, 1)
RETURN
END
To minimise the time taken to achieve a steady state condition, the start-up parameters have been selected, as far as possible, from the conditions selected for the main model logic.

Subroutine INITODS creates the order history file ORDHIST by running the order generator, subroutine ORDERS, for the equivalent of one year and summarising the orders received statistics by month. Thus, the basis for producing the orders received forecast, and subsequently the manufacturing plans, is consistent with the expected order input pattern.

It should be noted that the random number seeds are carried forward to the main model logic, thus selection of alternative seeds will change the start-up order history in detail.
SUBROUTINE INITODS(ORDHIST, TABLE5, TABLE6, TABLE7, ORDCOUNT, PIPELINE, ISEED4, ISEED5, ISEED6, ISEED7, AVGE4, CVARN4, IFILE, TABLE8, ISEED8, IPRINT)
INTEGER ORDHIST(5, 12), TABLE5(7, 2), TABLE6(7, 2), TABLE7(7, 2), ORDCOUNT, PIPELINE(20, 6, 4), IFILE(25, 23), TABLE8(2, 2)
DO 50 I=1, 4
DO 50 J=1, 12
50 ORDHIST(I, J)=0
DO 100 N=1, 12
DO 100 M=1, 4
CALL ORDERS(TABLE5, TABLE6, TABLE7, ORDCOUNT, PIPELINE, ISEED4, ISEED5, ISEED6, ISEED7, AVGE4, CVARN4, IFILE, TABLE8, ISEED8, IPRINT)
DO 100 I=1, 20
DO 100 J=2, 6
K=PIPELINE(I, J, 2)
IF(K,EQ:0)GO TO 100
ORDHIST(K, N)=ORDHIST(K, N)+PIPELINE(I, J, 3)
100 CONTINUE
RETURN
END
Subroutine ISSWIP is the mechanism for issuing components from stock to work in process. Facilities are provided to select items from the allocation file ALLQ in first-in-first-out priority or by due date, and to present the resultant material movement in familiar pick list format if required.

ISSWIP selects items sequentially from the allocation file ALLQ. The priority rule to be selected is determined by ISORT, the value of which is pre-defined for each simulation experiment. If ISORT is set to due date priority, ALLQ will be sorted into date sequence; first-in-first-out priority is achieved by by-passing the sort logic. It should be noted that the sort is by due date within product. The priority by product is determined by relative position in the part number master file, IFILE.

The quantity to be moved to work in process is determined by selecting an item from ALLQ. Sub-assemblies are moved as individual batches as defined in ALLQ and products are grouped by totalling all batches with a common shop order number.

When the allocation quantity has been determined, each component required to produce the assembly is checked for material availability. Items coded as "do not issue" are stored temporarily for subsequent checking at the next lowest level. If any parts are short, the next product or sub-assembly is selected from ALLQ.

Sub-assembly requirements that can be met from stock will be transferred from the allocation file to work in process. This is achieved by creating a new record in SUBQ, increasing the work in process quantity in IFILE and removing the record from ALLQ.

Equipment requirements that can be met from stock are identified by the shop order number. Each batch is transferred to work in process by adding a record to LINEQ, incrementing the work in process record in IFILE by the batch quantity and removing the item from ALLQ. The status of the order in OPFILE is amended by moving
the batch quantity from "in allocation" to "on-line". The cumulative quantity moved to work in process is incremented for subsequent use in the material issue procedure. This procedure avoids errors occurring if the volume of LINEQ is exceeded.

The total quantity moved for the product or sub-assembly is then processed by the issue list segment. If required, a listing of parts moved can be output in the form of a pick list. The physical stock and allocated stock of each component part is reduced by the quantity issued, including lower level components of "do not issue" assemblies.

The allocation file, ALLQ, is then consolidated by removing fully processed records.
Summarised Flow Chart - Subroutine Isswip
SUBROUTINE ISSWIP(IFILE, OPFILE, ALLQ, LINEQ, AFILE, IPRINT, REFWEEK,
1 SUBQ, ISORT, DIAGFILE, RUNTNE)
REAL AFILE(25,3)
INTEGER IFILE(25,23), OPFILE(125,5,5,10), ALLQ(12,50,6), LINEQ(5,50,
15), REQD, MFSK, HOLD(5,2), QUANTITY, DATED, DATEREQD, REFWEEK, SUBQ(50,4), 0
DIAGFILE(10,2), RUNTNE
IF(IPRINT.GT.1) GO TO 350
WRITE(2,320)
320 FORMAT(1HO,6X,** * *** ISSWIP ** * */))
350 CALL DATE(DATO)

C SORT ALLOCATION QUEUE BY DUE DATE

IF(ISORT.EQ.0) GO TO 3
DO 2 I=1,12
2 DO 1 J=1,49
DO 2 K=J+1,50
IF(ALLOCIK,41.EQ.0) GO TO 2
IF(ALLOCIJ,K,3).LE.ALLQ(I,K,3) GO TO 2
1 CONTINUE
2 CONTINUE

C SEARCH FOR ALLOCATIONS THAT ARE DUE FOR RELEASE

3 DO 400 N=1,12
2 IF(IFILE(N,11).GT.3) GO TO 400
2 IF(IFILE(N,11).EQ.0) GO TO 400
2 LEVEL=IFILE(N,11)
2 INDEX=0
2 DO 300 NP=1,50
2 IF(ALLOCN,NP,4).EQ.0) GO TO 300
2 IF(ALLOCN,NP,5).GT.0) GO TO 300
2 GO TO (4,6,6) LEVEL
300 CONTINUE
4 QUANTITY=0
4 IMAX=ALLQ(N,NP,6)
2 DO 5 L=NP,50
2 IF(ALLQ(N,L,6).NE.IMAX) GO TO 5
2 QUANTITY=QUANTITY+ALLQ(N,L,4)
5 CONTINUE
2 GO TO 7
6 QUANTITY=ALLQ(N,NP,4)
6 IMAX=0
7 IF(IPRINT.GT.2) GO TO 9
7 WRITE(2,8) AFILE(N), QUANTITY, IMAX
8 FORMAT(1HO,6X,'ISSUE DUE FOR RELEASE FOR PRODUCT',2X;A8,2X, 'QUANTITY', 16, 2X, 'SHOP ORDER', 16)
9 DATEREQD=ALLQ(N,NP,35+4)

C CHECK WHETHER ISSUE CAN BE MADE

NN=N
K=0
10 DO 60 I=1,9,2
   IF(IFILE(NN,1).EQ.0)GO TO 60
   I=I+1
   REQD=QUANTITY*IFILE(NN,IP)
   L=IFILE(NN,1)
   IF(IFILE(L,15).EQ.1)GO TO 50
   IF(REQD.LE.IFILE(L,16))GO TO 60
   INDEX=1
   IF(IPRINT.GT.2)GO TO 60
   WRITE(2,20)AFILE(L,1),REQD,IFILE(L,16)
   20 FORMAT(1HO,'SHORAGET OCCURRED ON ITEM',2X,A8,' WHERE QUANTITY REQUIRED = ',15,' PHYS STOCK = ',15)
   GO TO 60
   50 K=K+1
   HOLD(K,1)=L
   HOLD(K,2)=REQD
   60 CONTINUE
   IF(K.EQ.0) GO TO 100
   NN=HOLD(K,1)
   NN=HOLD(K,2)
   K=K-1
   GO TO 10
   100 IF(INDEX.EQ.1)GO TO 400
   MOVE STATUS TO W.I.P.

   SUB-ASSEMBLIES
   GO TO(167,155;155),LEVEL
   155 DO 157 L=1,50
      IF(SUBQ(L,3).EQ.0)GO TO 163
   157 CONTINUE
      IF(IPRINT.GT.5)GO TO 205
      WRITE(2,160)AFILE(N,1)
   160 FORMAT(1HO,'EXCEEDED SUBQ SIZE FOR PRODUCT NO.',A8)
      GO TO 205
   163 SUBQ(L,1)=N
      SUBQ(L,2)=ALLQ(N,NP,3)
      SUBQ(L,3)=ALLQ(N,NP,4)
      SUBQ(L,4)=3
      IFILE(N,18)=IFILE(N,18)+ALLQ(N,NP,4)
      ALLQ(N,NP,4)=0
   165 CONTINUE
      GO TO 105

   EQUIPMENTS
   167 QUANTITY=0
   DO 200 NR=1,50
      IF(ALLQ(N,NR,6).NE.IMAX)GO TO 200
   170 DO 175 L=1,50
      IF(LINEQ(N,L,74).EQ.0)GO TO 183
   175 CONTINUE
      IF(IPRINT.GT.5)GO TO 205
      WRITE(2,180)AFILE(N,1)
   180 FORMAT(1HO,'EXCEEDED LINE QUEUE SIZE FOR PRODUCT NO.',A8)
      GO TO 205
   183 JA=ALLQ(N,NR,71)
      JB=ALLQ(N,NR,72)
      JC=ALLQ(N,NR,73)
IF(ALLoN(N, NR, 4).LT.0) GO TO 187
DO 184 KEyNO=1,125
IF(OPFILE(KEYNO, JB, 1;10), EQ;JA) GO TO 186
184 CONTINUE
GO TO 195
186 DO 190 JJ=1,5
IF(OPFILE(KEYNO, JB, JJ;3), EQ;JC) GO TO 188
190 CONTINUE
GO TO 195
188 OPFILE(KEYNO; JB, JJ, 7)=ALLQ(N; NR, 4) * OPFILE(KEYNO, JB, JJ, 7)
OPFILE(KEYNO, JB, JJ, 6)=OPFILE(KEYNO, JB, JJ, 6) - ALLQ(N, NR, 4)
187 QUANTITY=QUANTITY+ALLQ(N; NR, 4)
DO 189 JJ=1,4
189 LINEQ(N, L;7)=ALLQ(N; NR, 4)
LINEQ(N, L, 5)=3
IFILE(N;18)=IFILE(N;18)+ALLQ(N, NR, 4)
ALLQ(N, NR, 4)=0
GO TO 200
195 IF(1PRINT.GT.5) GO TO 200
WRITE(2;Z50)JA; JS; JC; OPFILE(KEVNO, JB; JJ, 3), REFWEEK
250 FORMAT(1HO,6X,'NO ENTRY FOUND IN OPFILE FOR ORDER',14,' ITEM',14,'DUE WEEK',14/6X,'ACTUAL WEEK ',14,6X;'REFWEEK ',14)
1PRINT=1
DO 255 JJ=1,10
255 IFPRINT=IFPRINT+12=1,10
RUNTIME=RUNTIME+12=1,10
GO TO 200
200 CONTINUE

PRODUCE ISSUE LIST AND REDUCE PHYSICAL STOCK

105 K=0
110 IF(1PRINT.GT.3) GO TO 115
WRITE(2,500)DATEREQD, AFILE(N, 1), QUANTITY
500 FORMAT(1HO,6X,'ISSUE DATE:','A8/6X,'WEEK REQUIRED:','12//,23X,'**'
1***/6X,'ASSEMBLY NUMBER '+,'A8,'**','6X,'QUANTITY REQUESTED:','I3
2/23X,'**/**/**/6X,'PART NUMBER ','4X,'DESCRIPTION ','4X,'REQUIRED'
3')
115 DO 150 =1,9;2
IF(IFILE(N, IP), EQ, 0) GO TO 150
IP=IP+1
REQD=QUANTITY*IFILE(N, IP)
L=IFILE(N, 1)
150 CONTINUE
IF(IFILE(L;15), EQ, 1) GO TO 130
IFILE(L;16)=IFILE(L;16)-REQD
IFILE(L;17)=IFILE(L;17)-REQD
IF(IPRINT.GT.3) GO TO 150
WRITE(2,120) (AFIPLE(L;J), J=1,3), REQD
120 FORMAT(1HO,3X,'A8','4X,'2A8,16)
GO TO 150
130 IF(1PRINT.GT.3) GO TO 145
WRITE(2,140) AFILE(L;1)
140 FORMAT(1HO,3X,'A8','4X,'SEPARATE ISSUE LIST * * * * *')
145 K=K+1
HOLD(K,1)=L
HOLD(K,2)=REQD
150 CONTINUE
IF(K, EQ, 0) GO TO 300
N=HOLD(K,1)
QUANTITY=HOLD(K,2)
K=K-1
GO TO 110
300 CONTINUE
205 DO 220 K=1,49
   IF(ALLQ(N,K,4).GT.0)GO TO 220
   DO 210 M=K+1,50
      IF(ALLQ(N,M,4).GT.0)GO TO 215
   CONTINUE
   GO TO 400
   215 DO 218 L=1,6
      ALLQ(N,K,L)=ALLQ(N,M,L)
      ALLQ(N,M,L)=0
   CONTINUE
218 CONTINUE
220 CONTINUE
400 CONTINUE
RETURN
END
The subroutine QPLAN requires, as part of the decision process, constraints within which changes can be effected. These constraints, or limits, may be labour or material related and, dependent upon the time horizon, may be a maximum or a minimum parameter.

The labour constraint simulates the effect of increasing the planned requirement and represents the maximum plan in any one quarter with reference to the previous quarter's activity.

The labour related rules incorporated into LIMIT are:

* No increase in activity permissible in the current quarter.
* If the previous quarter was zero, a maximum of 250 is permissible.
* If the previous quarter up to 150, a maximum of 3 times the previous plan is permitted.
* If the previous quarter up to 500, a maximum of 2.5 times the previous plan is permitted.
* If the previous quarter up to 2000, a maximum of 2.0 times the previous plan is permitted.
* If the previous quarter more than 2000, a maximum of 1.5 times the previous plan is permitted.
* It is assumed that there are no labour constraints on reduction in planned requirements.

Material constraints have been calculated as a maximum and a minimum, and both are related to the lead time available before the change is required and the likely success in achieving the change within the ability of the suppliers to respond to the requests. It should be noted that the parameters chosen do not necessarily reflect the aggregate internal and external lead times (or response times), either in the model or the real world, but are similar to those currently used in the manual plan derivation.
The rules are summarised in the table below:

<table>
<thead>
<tr>
<th>PERIOD</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXIMUM PLAN</td>
<td>Px100%</td>
<td>Px100%</td>
<td>*Px150%</td>
<td>No limit or 150</td>
</tr>
<tr>
<td>MINIMUM PLAN</td>
<td>Px100%</td>
<td>Px100%</td>
<td>Px50%</td>
<td>0</td>
</tr>
</tbody>
</table>

* whichever is the higher

From period 4 onwards there is considered to be no material constraint (either maximum or minimum), thus the labour constraint becomes the overriding factor.

In arriving at the maximum plan, the lower of the labour or material constraint is used.

Although the labour constraint is strictly an "off-line" parameter, for simplicity the maximum and minimum are deemed to apply to the material, or on-line plan.
Summarised Flow Chart - Subroutine Limit
SUBROUTINE LIMIT(PLAN, QTR, MATPLN, PLNMAX, PLNMIN, CUMDIF)
INTEGER PLAN, LABMAX, QTR, MATMAX, MATPLN, PLNMAX, PLNMIN, CUMDIF

SET MINIMUM LIMIT
GO TO (230, 10, 20, 30, 40, 40, 40, 50) QTR
10 PLNMIN = MATPLN
GO TO 50
20 PLNMIN = MATPLN
GO TO 50
30 PLNMIN = MATPLN * 0.5
GO TO 50
40 PLNMIN = 0

SET LABOUR CONSTRAINT
IF (QTR = 2) 45, 45, 50
45 LABMAX = PLAN
GO TO 140
50 IF (PLAN = 60) 60, 60, 70
60 LABMAX = 250
GO TO 140
70 IF (PLAN - 150) 80, 80, 90
80 LABMAX = PLAN * 3
   IF (LABMAX = 250) 60, 140, 140
90 IF (PLAN = 500) 100, 100, 110
100 LABMAX = PLAN * 2.5
   GO TO 140
110 IF (PLAN = 2000) 120, 120, 130
120 LABMAX = PLAN * 2.0
   GO TO 140
130 LABMAX = PLAN * 1.5

SET MAXIMUM MATERIAL LIMITS
140 GO TO (230, 150, 160, 170, 190, 190, 190, 190) QTR
150 MATMAX = MATPLN
GO TO 200
160 MATMAX = MATPLN + 1.25 + CUMDIF
   GO TO 200
170 MATMAX = MATPLN * 1.75
   IF (MATMAX < 150) 180, 180, 200
180 MATMAX = 150
   GO TO 200
190 MATMAX = LABMAX
200 IF (MATMAX < LABMAX) 210, 210, 220
210 PLNMAX = MATMAX
   GO TO 224
220 PLNMAX = LABMAX
224 IF (PLNMIN < PLNMAX) 228, 228, 228
226 PLNMIN = PLNMAX
228 RETURN
230 RETURN 1
END
B16 - SUBROUTINE LOAD

Customer orders generated by subroutine ORDERS are entered in file PIPELINE in preparation for order promising (assignment of due dates), also known as order loading. Subroutine LOAD performs the task of due date assignment and subsequently creates corresponding records in the order book, ORDBK and the orders placed file, OPFILE.

Orders are extracted sequentially from file PIPELINE and the next available space in OPFILE is assigned. The PIPELINE record will identify the order reference, the number of items and whether part shipment is permissible. For "no part shipment" orders, the longest delivery lead time is determined by searching for the item with the latest delivery date, the available capacity being the difference between the committed load as contained in file TOTAL and the delivery plan in DELPLAN. If insufficient capacity is available to support the order, a 26 week delivery lead time is selected on the assumption that the next re-planning cycle will accommodate the overload. Order line items are committed to the delivery plan by searching for the first "available to promise" quantity (difference between the delivery plan and total committed load) beyond the minimum loading date. The minimum loading date is either the minimum order book parameter (OCFMIN) for "part shipment" orders, or the maximum lead time determined as above for "no part shipment" orders.

A successful search will cause the order to be committed to the plan, in which case the total load for the appropriate week number will be incremented by the quantity loaded. The quantity selected is either the full item quantity or the available to promise, whichever is the smaller. If the full quantity cannot be loaded, the balance will be carried forward for loading in subsequent periods, up to a maximum of five periods in total. In the interests of file economy, quantities that cannot be accommodated over five periods cause the remainder of the order quantity to be truncated.
If insufficient delivery plan is available, a 26 week delivery lead time is selected for the full item quantity.

Each quantity loaded will cause a corresponding record to be created in the order book (ORDBK) for the appropriate product and week number, and the orders placed file (OPFILE). The delivery lead time achieved is registered in file LEADTM for subsequent analysis if required.
Summarised Flow Chart - Subroutine Load

A
B
C

D
REDUCE QUANTITY TO LOAD

QUANTITY LEFT TO LOAD?

N
Y

INSUFFICIENT PLAN?

Y
LOAD ON 26 WEEK LEAD TIME

N
ADD TO ORDBK & OPFILE

ADD TO TOTAL LOAD

RETURN

E
SUBROUTINE LOAD(DELPLAN, OCFMIN, WEEK, TOTAL, PIPELINE, ORDBK, OPFILE, ORDERHIST, IPRINT, FILE, ORDCOUNT, REFWK, LEADTM)
REAL FILE(25, 3)
INTEGER DELPLAN(5, 52), OCFMIN, WEEK, TOTAL(5, 52), PIPELINE(20, 6, 4), 0 RDBK(5, 52, 20, 5), OPFILE(125, 5, 5, 10), ORDERHIST(5, 12), IPRINT, ORDN0, IT EMCOUNT, TYPE, QTY, ORDCOUNT, FILE(25, 23), REFWK, ACTWK, LEADTM(5, 52)
IF(IPRINT.GT.1) GO TO 7
WRITE(2, 5)
5 FORMAT(1HO, 6X, 'LOAD *** *** *** *** *** ***'),
7 JZ = 0
WEEK = OCFMIN - 4
10 JZ = JZ + 1
IF(PIPELINE(JZ, 1, 1), EQ, 0) GO TO 150
C FIND NEXT SPACE IN OPFILE
DO 13 KEYNO = 1, 125
13 CONTINUE
IF(IPRINT.GT.5) GO TO 160
WRITE(2, 15)
15 FORMAT(1HO, 6X, 'EXCEEDED OPFILE SIZE'),
160 ORDN0 = PIPELINE(JZ, 1, 1)
ITEMCOUNT = PIPELINE(JZ, 1, 3)
IF(IPRINT.GT.3) GO TO 25
WRITE(2, 20) ORDN0
20 FORMAT(1HO, 12X, 'ORDER NUMBER', l6, 2X, 'ITEM', 6X, 'PROD', 2X, 'QTY'),
25 IF(PIPELINE(JZ, 1, 2)
MAXWEEK = 1
GO TO(35, 26)
C FIND LONGEST LEAD TIME FOR NO PART SHIPMENT
26 DO 30 J = 1, ITEMCOUNT
L = J + 1
TYPE = PIPELINE(JZ, L, 2)
27 CONTINUE
IF(DELPLAN(TYPE, I), GT, TOTAL(TYPE, I)) GO TO 29
29 IF(L, LE, MAXWEEK, GO TO 30
30 MAXWEEK = 1
31 CONTINUE
33 IF(IPRINT.GT.2) GO TO 35
WRITE(2, 30) ORDN0, MAXWEEK
30 FORMAT(1HO, 6X, 'NO PART SHIPMENT FOR ORDER NUMBER', l6, 6X, 'LONGEST LEAD TIME LOADED WEEK'),
35 IF(MAXWEEK.GT.5) GO TO 39
MAXWEEK = N
39 DO 110 JM = 1, ITEMCOUNT
L = JM + 1
ITEM = PIPELINE(JZ, L, 1)
TYPE = PIPELINE(JZ, L, 2)
QTY = PIPELINE(JZ, L, 3)
IF(IPRINT.GT.2) GO TO 50
WRITE(2,40) ITEM, AFILE(TYPE,1), QTY
40 FORMAT(1HO, 32X, I4, 2X, A8, 2X, I4)
50 ORDHIST(TYPE,12)=ORDHIST(TYPE,12)+QTY
M=0
DO 100 I=MAXWEEK, 52
IF(QTY.LE.(DELPLAN(TYPE,1)-TOTAL(TYPE,1))) GO TO 80
IF(DELPLAN(TYPE,1).LE.TOTAL(TYPE,1)) GO TO 100
NUMBER=DELPLAN(TYPE,1)-TOTAL(TYPE,1)
GO TO 70
80 NUMBER=QTY
70 DO 60 J=1,20
IF(ORDBK(TYPE,J,1),EQ,0) GO TO 90
60 CONTINUE
IF(IPRINT.GT.5) GO TO 100
IZ=REFWEEK+1
WRITE(2,65) AFILE(TYPE,1), IZ
65 FORMAT(1HO, 6X, 'EXCEEDED ORDER BOOK SIZE FOR PRODUCT', 2X, A8, 2X, 'IN
1 WEEK NUMBER',I6)
GO TO 100
90 QTY=QTY-NUMBER
92 DO 95 N=1,5
IF(OFFILE(KEYNO, ITEM,M,1),EQ,0) GO TO 99
95 CONTINUE
IF(IPRINT.GT.2) GO TO 110
WRITE(2,400)
400 FORMAT(1HO, 6X, 'NUMBER OF WEEKS EXCEEDS FIVE, QUANTITY TRUNCATED')
GO TO 110
99 ORDBK(TYPE,1,J,1)=ORDNO
ORDBK(TYPE,1,J,2)=ITEM
ORDBK(TYPE,1,J,3)=NUMBER
ORDBK(TYPE,1,J,4)=0
ORDBK(TYPE,1,J,5)=PIPELINE(JZ,1,2)
ACTWK=REFWEEK+1
OFFILE(KEYNO, ITEM,M,1)=TYPE
OFFILE(KEYNO, ITEM,M,2)=NUMBER
OFFILE(KEYNO, ITEM,M,3)=ACTWK
OFFILE(KEYNO, ITEM,M,4)=NUMBER
OFFILE(KEYNO, ITEM,M,10)=ORDNO
LEADTM(TYPE,1-WEEK)=LEADTM(TYPE,1-WEEK)+NUMBER
TOTAL(TYPE,1)=TOTAL(TYPE,1)+NUMBER
IF(QTY.EQ.0) GO TO 101
100 CONTINUE
IF(IPRINT.GT.2) GO TO 101
WRITE(2,600) AFILE(TYPE,1), ORDNO
600 FORMAT(1HO, 6X, 'EXHAUSTED AVAILABLE DELIVERY PLAN FOR PRODUCT NUMBE
1R1, 2X, A8, 2X, 'ORDER NUMBER', I15, ' LOADED ON 26 WEEK LEAD TIME')
IF DELIVERY PLAN IS INSUFFICIENT, ORDER LOADED ON 26 WEEK LEAD TIME
101 JL=WEKKE+26
IF(N, EQ, 5) GO TO 110
M=N+1
102 DO 103 J=1,20
IF(ORDBK(TYPE,JL,J,1),EQ,0) GO TO 105
103 CONTINUE
IF(IPRINT.GT.5) GO TO 104
WRITE(2,65) AFILE(TYPE,1), I1
104 JL=JL+1
IF(JL,GT,52) GO TO 110
GO TO 102
105 ORDBK(TYPE, JL, J, 1) = ORDN0
   ORDBK(TYPE, JL, J, 2) = ITEM
   ORDBK(TYPE, JL, J, 3) = QTY
   ORDBK(TYPE, JL, J, 4) = 0
   ORDBK(TYPE, JL, J, 5) = PIPELINE(JZ, 1, 2)
   ACTWK = REFWEEK + JL
   OPFILE(KEYNO;ITEM;M;1) = TYPE
   OPFILE(KEYNO;ITEM;M;2) = QTY
   OPFILE(KEYNO;ITEM;M;3) = ACTWK
   OPFILE(KEYNO;ITEM;M;4) = QTY
   OPFILE(KEYNO;ITEM;M;10) = ORNO
   LEADTH(TYPE, JL = WEEK) = LEADTM(TYPE, JL = WEEK) + QTY
   TOTAL(TYPE, JL) = TOTAL(TYPE, JL) + QTY
110 CONTINUE
   GO TO 10
150 IF (PRINT .GT. 3) GO TO 160
   CALL ORDBKPRT(ORDBK, AFILE, IFILE, TOTAL, REFWEEK)
   CALL OPFPRT(OPFILE, AFILE)
160 RETURN
END

<table>
<thead>
<tr>
<th>ORDER NUMBER</th>
<th>16</th>
<th>ITEM</th>
<th>PROD</th>
<th>QTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO PART SHIPMENT FOR ORDER NUMBER 16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LONGEST LEAD TIME LOADED WEEK 14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MF6AM</td>
<td>02</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MF6AM</td>
<td>02</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MF6AM</td>
<td>01</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>AC15PU</td>
<td></td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>AC15PU</td>
<td></td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ORDER NUMBER</th>
<th>17</th>
<th>ITEM</th>
<th>PROD</th>
<th>QTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MF6AM</td>
<td>01</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MF6AM</td>
<td>01</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Sample Order Loading Report
Under ideal conditions, which implies that orders are received as forecast and deliveries take place according to plan, the delivery plan can be used as a basis for order promising. When conditions are less than ideal, it is often desirable to promise orders at a different level, to accommodate for example

- excess finished equipment stock
- insufficient finished equipment stock
- excessive overdue orders
- poor performance to plan

The latter is not addressed by the model since it is normally evident by either excessive overdue orders or insufficient finished equipment stock.

Subroutine LOADPLAN is used to modify the delivery plan derived from subroutine WEEKLYPLAN to reflect the above conditions.

On entering subroutine LOADPLAN, the modeller has the opportunity to by-pass the subroutine logic by setting ILOAD to zero. This will cause the delivery plan, DELPLAN, to be written directly into the loading plan, NETPLAN.

Setting ILOAD to "1" causes the delivery plan to be modified according to the level of overdue orders and excess stock. The overdue load is derived by searching the order book file, ORDBK, for the appropriate product for outstanding quantities which either have the overdue flag set or are overdue in the current quarter.

Excess stock is the difference between the actual free stock (physical stock less allocated stock) and one half of the maximum authorised stock, all of which are contained within IFILE.

The net result of the overdue orders and excess stock is projected to the minimum delivery lead time by comparing the difference between the delivery plan (DELPLAN) and the committed load (TOTAL) over the period.
The resultant quantity is then used to modify the delivery plan. A negative value indicates an overload condition and a positive value suggests excess capacity.

Overload conditions cause the loading plan to be reduced in the earliest available week after the minimum loading period, to a minimum determined by the existing committed load. Excess capacity is accommodated by increasing the loading plan by a factor of 15% beyond the minimum loading period until the excess capacity has been accounted for. This simulates the application of overtime working in the test departments (this condition is normally as a result of excess stock).

The resultant modified delivery plan is written into file NETPLAN for subsequent application in subroutine LOAD.
Subroutine Loadplan

Bypass required?

Y

N

Establish calculation parameters

Select next product

Calculate overdue load

Calculate excess stock

Calculate nett result at minimum loading week

Adjust loading plan

RETURN

Summarised Flow Chart - Subroutine Loadplan
SUBROUTINE LOADPLAN(DELPLAN, NETPLAN, FILE, IPRINT, WEEK, OCFMIN, ORDERBK, TOTAL)
REAL FILE(25, 3)
INTEGER DELPLAN(5,52), NETPLAN(5,52), FILE(25, 23), WEEK, OCFMIN, XSTOCK, ORDERBK(5,52, 20), TOTAL(5,52)

TO BYPASS LOADPLAN SET ILOAD TO 101

ILOAD = 1
IF(IPRINT.GT.1) GO TO 2
WRITE(2,1)
1 FORMAT(1HO, 6X, 'LOADPLAN')
2 IF(ILOAD.EQ.1) GO TO 4
DO 3 I = 1, 5
DO 3 J = 1, 52
NETPLAN(I,J) = DELPLAN(I,J)
3 CONTINUE
GO TO 150
4 N = WEEK + OCFMIN - 1
JA = WEEK
JB = WEEK + 19
IF(IPRINT.GT.2) GO TO 5
WRITE(2, 200)
200 FORMAT(1HO, 6X, 'PRODUCT',6X, 'OVERDUES', 6X, 'EXCESS STOCK', 6X, 'NET LOAD')
WRITE(2, 250)(J; J = JA, JB)
250 FORMAT(1HO, 20X,'2014')

CALCULATE OVERDUE LOAD

DO 100 I = 1, 5
IF(FILE(I, 11).NE. 1) GO TO 100
NETLOAD = 0
DO 90 J = 1, 52
DO 90 K = 1, 20
IF(ORDERBK(I, J, K).EQ. 0) GO TO 90
NETLOAD = NETLOAD - ORDERBK(I, J, K)
90 CONTINUE
IF(WEEK.EQ. 1) GO TO 30
DO 20 J = 1, WEEK + 1
NETPLAN(I, J) = 0
DO 20 K = 1, 15
IF(ORDERBK(I, J, K).EQ. 1) GO TO 20
NETLOAD = NETLOAD - ORDERBK(I, J, K)
20 CONTINUE
30 IF(IPRINT.GT.2) GO TO 40
WRITE(2, 300) NETLOAD
300 FORMAT(1HO, 18X,'16')

CALCULATE EXCESS STOCK

XSTOCK = FILE(I, 16) - FILE(I, 23) / 2.0
NETLOAD = NETLOAD + XSTOCK

CALCULATE LOAD DIFFERENCE AT OCFMIN

DO 50 J = WEEK + N
NETPLAN(I, J) = DELPLAN(I, J)
50 NETLOAD = NETLOAD + DELPLAN(I, J) - TOTAL(I, J)
IF(IPRINT.GT.2)GO TO 60  
WRITE(2;400)XSTOCK, NETLOAD  
400 FORMAT(1H+,35X;16;10X;16)  
C  
RESET LOADING PLAN  
  60 IF(NETLOAD.LE.0)GO TO 80  
  DO 70 J=N+1,52  
  NETPLAN(I,J)=DELPLAN(I,J)*1.15  
  NETLOAD=NETLOAD-NETPLAN(I,J)+DELPLAN(I,J)  
  IF(NETLOAD.LE.0)GO TO 90  
  70 CONTINUE  
  GO TO 90  
  80 DO 87 J=N+1,52  
  IF((TOTAL(I,J)+DELPLAN(I,J)).GT.NETLOAD)GO TO 85  
  83 NETPLAN(I,J)=DELPLAN(I,J)+NETLOAD  
  GO TO 90  
  85 NETPLAN(I,J)=TOTAL(I,J)+DELPLAN(I,J)-TOTAL(I,J)  
  87 CONTINUE  
  90 IF(J.GT.5)GO TO 95  
  DO 92 M=J+1,52  
  92 NETPLAN(I,M)=DELPLAN(I,M)  
  95 IF(IPRINT.GT.4)GO TO 100  
  WRITE(2,500)FILENAME(I;),NETPLAN(I,J);J=JA,JB),(NETPLAN(I,J),J=JA,JB)  
  500 FORMAT(1H0,6X,A8,5X,'DELPLAN',1X,2X,2014//20X,'NETPLAN',2X,2014)  
  100 CONTINUE  
  150 RETURN  
END
Subroutine MONTHEVENT controls the activities which are required to be executed each month, all of which are independent modules.

**SUBPROG** - establishes the sub-assembly manufacturing programme from the requirements data and modifiers (EBQ, stock levels).

**OHRESET** - resets the orders received history file in preparation for the acceptance of new data.

**STOCKVAL** - calculates and displays the value of stock in each category.

**PNMFPRINT** - optional module displaying the status of the item master data.

**LOADPLAN** - establishes the plan to be used for order loading.

Two counters are maintained within subroutine MONTHEVENT; the first establishes the actual simulation month and the second resets the decremental counter which subsequently determines the time of activation of MONTHEVENT.
CALL MONTHEVENT

DETERMINE SUB-ASSEMBLY PROGRAMME

RESET ORDER HISTORY FILE

CALCULATE STOCK VALUATION

DIAGNOSTIC LEVEL > 3

PRINT ITEM MASTER

ESTABLISH ORDER LEADING PLAN

RESET MONTH COUNTER

RETURN

SUMMARISED FLOW CHART - SUBROUTINE MONTHEVENT
SUBROUTINE MONTHEVENT(AFILe, IFILE, REQFILE, SUBFILE, WEEK, ORDHIST, IPRINT, MONTH, OPFILE, DELPLAN, NETPLAN, OCFMIN, ORDBK, ALLQ, TOTAL)
REAL AFILe(25,3)
INTEGER IFILE(25,23), REQFILE(25,52), WEEK, SUBFILE(12,26), ORDHIST(5,12), OPFILE(125,5,5,10), DELPLAN(5,52), NETPLAN(5,52), OCFMIN, ORDBK(25,52,20,5), ALLQ(12,50,6), TOTAL(5,52)
IF(IPRINT.GT.1)GO TO 20
WRITE(2,10)
10 FORMAT(1HO,* * * * MONTHEVENT * * * * )
20 CALL SUBPROG(REQFILE, WEEK, IFILE, IPRINT, SUBFILE, AFILe, ALLQ)
   CALL OHRESET(ORDHIST)
   CALL STOCKVAL(AFILe, MONTH, OPFILE)
   IF(IPRINT.GT.3)GO TO 40
   CALL PNMPPRINT(AFILe, IFILE)
40 CALL LOADPLAN(DELPLAN, NETPLAN, IFILE, IPRINT, WEEK, OCFMIN, ORDBK, AFILe)
   TOTAL)
   MONTH=MONTH+1
RETURN
END
The subroutine NETTING is used to adjust, or "net", the requirements for each item for free stock and planned buffer (safety) stocks.

The requirements for the item are presented to the programme as a weekly plan over a 52 week horizon. The planned buffer stock is maintained on the PNMF in terms of weeks worth of stock and the opening value of free stock is also derived from the PNMF.

Each month, the planned level of buffer stock at the end of the month is calculated by summing the next "x" weeks of plan, where "x" is the number of weeks worth of buffer required. A net quantity is derived, being the difference between the previous buffer (or actual stock in the first period) and the revised buffer. The resultant net quantity is then offset against the gross requirements plan to provide a net requirements plan.

Two examples of the logic are shown below:

<table>
<thead>
<tr>
<th>WEEK</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross Plan</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Planned buffer (4 weeks)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>60*</td>
</tr>
<tr>
<td>Opening stock</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net Value</td>
<td>+20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net Plan</td>
<td>30</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>25</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

*Assumes constant 15 pw
<table>
<thead>
<tr>
<th>WEEK</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross Plan</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>25</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Planned Buffer</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>25</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Opening Stock</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net Value</td>
<td>+20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-25</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Net Plan</td>
<td>20</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**SUBROUTINE NETTING**

**SELECT NEXT PERIOD**

**CALCULATE REQUIRED BUFFER**

**CALCULATE NETT ADJUSTMENT**

**ADJUST PLAN**

**RETURN**

---

**Summarised Flow Chart - Subroutine Netting**
SUBROUTINE NETTING(BUFFERWKS, STOCK, WORKFILE)
  INTEGER BUFFERWKS, STOCK, NETQTY, OLDBUFFER, WORKFILE(52)

  OLDBUFFER=STOCK
  DO 100 I=1,49,4
  IF(BUFFERWKS.GT.0) GO TO 5
  BUFFER=0
  GO TO 30

  CALCULATE BUFFER FOR NEXT PERIOD

  5 BUFFER=0
     J=I+BUFFERWKS-1
     DO 10 L=1+4, J+4
        IF(L.GT.52) GO TO 30
        BUFFER=BUFFER+WORKFILE(L)
     10 CONTINUE

  CALCULATE NET MOVEMENT OF BUFFER STOCK

  30 NETQTY=BUFFER-OLDBUFFER

  ADJUST PLAN

     J=I+3
     DO 60 K=1, J
        IF(WORKFILE(K).NE.NETQTY) 40,50,50
     40 NETQTY=NETQTY+WORKFILE(K)
        WORKFILE(K)=0
        GO TO 60
     50 WORKFILE(K)=WORKFILE(K)*NETQTY
        NETQTY=0
     60 CONTINUE

  OLDBUFFER=BUFFER-NETQTY

  100 CONTINUE
  RETURN
END
Subroutine NORMAL is a general purpose facility for generating normally distributed random variates. The subroutine is presented with the mean and co-efficient of variance of the normal distribution. A seed is also provided for use by the random number generator RANDOM.

The normal variate generator is based on the Central Limit Theorem, which states that the probability distribution of the sum of $N$ independently and identically distributed random variates $x_i$ with respective means $\mu_i$ and variances $\sigma_i^2$, as $N$ becomes very large, approaches the normal distribution asymptotically. The resultant distribution has mean and variance:

$$\mu = \sum_{i=1}^{N} \mu_i$$
$$\sigma^2 = \sum_{i=1}^{N} \sigma_i^2$$

The probability density function $f(x)$ for a normal distribution is given by:

$$f(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{x - \mu_x}{\sigma_x} \right)^2}$$

for $-\infty < x < \infty$

If $\mu_x = 0$ and $\sigma_x = 1$, the distribution function is known as the "standard normal distribution", with density function

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} z^2}$$

for $-\infty < z < \infty$

To convert a normal distribution into the standard form

$$z = \frac{x - \mu_x}{\sigma_x}$$
If we now consider the Central Limit Theorem, where \( r_1, r_2, \ldots, r_N \) are independent random variables each having the same probability distribution with expected value \( E(r_i) = \theta \) and variance \( \text{Var}(r_i) = \sigma^2 \), then

\[
\lim_{N \to \infty} P \left[ \frac{\sum_{i=1}^{N} r_i - N\theta}{\sqrt{N\sigma^2}} < b \right] = \frac{1}{\sqrt{2\pi}} \int_{a}^{b} e^{-\frac{1}{2}z^2} \, dz
\]

Where

\[
E \left( \sum_{i=1}^{N} r_i \right) = N\theta \\
\text{Var} \left( \sum_{i=1}^{N} r_i \right) = N\sigma^2 \\
z = \frac{\sum_{i=1}^{N} r_i - N\theta}{\sigma\sqrt{N}}
\]

If we now consider the uniformly distributed random variate, the expected value and variance are given by

\[
E(x) = \int_{a}^{b} \frac{1}{b-a} x \cdot dx = \frac{a+b}{2}
\]

\[
\text{Var}(x) = \int_{a}^{b} \frac{(x - E(x))^2}{b-a} \cdot dx = \frac{(b-a)^2}{12}
\]

If we assume that the intervals \( a, b \) are 0, 1 respectively, the sum of \( K \) uniformly distributed random variates \( r_1, r_2, \ldots, r_K \) (where \( r_i \) is defined over the interval \( 0 \leq r_i \leq 1 \)).

\[
\theta = \frac{0 + 1}{2} = 1/2
\]

\[
\sigma = \frac{b - a}{\sqrt{12}} = \frac{1}{\sqrt{12}}
\]
\[ z = \frac{\sum_{i=1}^{K} r_i - \frac{K}{2}}{\sqrt{K/12}} \]

Since \( z \) is the standard normal deviate, then

\[ \frac{x - \mu_x}{\sigma_x} = \frac{\sum_{i=1}^{K} r_i - \frac{K}{2}}{\sqrt{K/12}} \]

or \( x = \sigma_x \left( \frac{12}{K} \right)^{1/2} \left( \sum_{i=1}^{K} r_i - \frac{K}{2} \right) + \mu_x \]

This now provides the basis for generating random normal variates given a series of \( K \) uniformly distributed random variates, and the defined mean \( (\mu_x) \) and standard deviation \( (\sigma_x) \) of the desired normal distribution.

The larger the value of \( K \), the more accurate is the result, especially at the distribution tails. However, a value of \( K = 12 \) simplifies the computation and provides reasonably accurate results up to three standard deviations.

Thus, if \( K = 12 \)

\[ x = \sigma_x \left( \sum_{i=1}^{12} r_i - 6 \right) + \mu_x \]  

The construction of subroutine NORMAL is therefore straightforward. Given the definition of the normal distribution \( (\sigma_x, \mu_x) \), twelve uniformly distributed random variates are derived from subroutine RANDOM. These are presented to equation (1) above, and the result "\( x \)" is returned to the calling programme.
NORMAL

SUBROUTINE NORMAL(AVGE, X, J, CVARN)

SUM = 0.0
SVAR = CVARN * AVGE
DO 10 I = 1, J
10 CALL RANDOM(R, J)
SUM = SUM + R
X = SVAR * (SUM - 6.0) + AVGE
RETURN
END
The off-line point is taken to be the movement between the assembly work-in-progress queue to either test or stock, dependent upon the type of item. Sub-assemblies and equipment stock orders will be moved to stock; customer orders will move directly to the test work-in-progress queue.

Assembly work-in-progress queues are held in files LINEQ for equipments and SUBQ for sub-assemblies. Subroutine OFFLINE offers the facility to sort each queue into due date sequence or default to a first-in-first-out priority, in which case the sort segments are by-passed. Each queue represents a logically independent manufacturing department, and no interaction between each queue is assumed.

The sub-assembly queue, SUBQ, represents the work-in-progress in the sub-assembly department, the manufacturing capacity of which is determined by the gross quarterly capacity requirement calculation in sub-routine EBQ. The capacity is measured in total sub-assemblies per quarter and must first be converted into the equivalent rate per week.

Items are selected sequentially from the sub-assembly queue and compared with the available capacity. Items cannot be moved unless the timer is set at zero or less, recognising the minimum throughput time determined in STKALL. If sufficient capacity is available, the item quantity will be removed from the queue and the item physical stock is augmented in IFILE. The capacity balance is then reduced by the quantity moved and the next item selected from SUBQ. The process is continued until the capacity available for the week has been consumed or the work-in-progress queue is exhausted. SUBQ records which have been fully processed (zero quantity outstanding) are removed and the file consolidated.

Equipments are selected from each product queue and matched against the manufacturing rate as defined byREQFILE for the appropriate week. The processing logic is dependent upon the
LINEQ record being either a stock order or a customer order. Stock orders will pass directly to stock on leaving LINEQ for subsequent allocation to customer orders in subroutine FESALL. Customer orders pass directly from LINEQ to the test work-in-progress, TESTQ.

For each equipment type, LINEQ is accessed sequentially and the order records with the time set at zero or less are selected. The quantity moved for each order is the order line quantity or the balance to be moved off-line as defined by REQFILE for the appropriate week. The orders placed file, OPFILE, is searched to locate the corresponding order record, and a successful match will cause the status to change from "on-line" to "test". The LINEQ data will be transferred to the first available record in TESTQ for the appropriate product and the time set to the minimum throughput time in test.

Stock orders will by-pass the orders placed file transaction and will augment the physical stock record in IFILE instead of moving to TESTQ. Both stock and customer order will cause the line work-in-progress record in IFILE, the LINEQ record and the off-line plan balance in REQFILE to be decremented by the quantity moved.

Orders will continue to be selected from LINEQ until the off-line plan is consumed or the eligible work-in-progress is exhausted. LINEQ records which have been fully processed are removed and the file consolidated.
SUBROUTINE OFFLINE

PRIORITY RULE?

FIFO

SELECT NEXT ITEM IN SUBQ

SUB-ASSY CAPACITY LEFT?

MOVE ITEM TO STOCK

REDUCE CAPACITY

REORGANISE SUBQ

SELECT NEXT PRODUCT

SELECT MIX FROM LINEQ

ESTABLISH MAKE QUANTITY

Y

STOCK ORDER

INC an C Increase stock

N

FIND SPACE IN TESTQ

TESTQ FULL?

PRINT TESTQ FULL

Cont. over.
Summarised Flow Chart - Subroutine Off-line
SUBROUTINE OFFLINE(LINEQ, TESTQ, OPFILE, WEEK, REQFILE, IFILE, IPRINT, AF
1LE, REFQUEEK, CAPACITY, ISORT, SUBQ, DIAGFILE, RUNTIME)
REAL AFILE(25, 3)
INTEGER LINEQ(5, 50, 5), TESTQ(5, 50, 5), OPFILE(125, 5, 5, 10), WEEK, REQFI
1LE(25, 52), IFILE(25, 23), REFWEKEK, CAPACITY, WKCAP, SUBQ(50, 4), QTY, DI
2AGFILE(10, 2), RUNTIME
WKCAP = CAPACITY / 13.0
IF(IPRINT.GT.1) GO TO 1
WRITE(2, 560)
560 FORMAT(1HO, 6X, 'OFFLINE ')
WRITE(2, 570) WKCAP
570 FORMAT(1HO, 6X, 'WKCAP = 'I6)

SORT SUB QUEUE
1 IF(ISORT.EQ.0) GO TO 5
   DO 2 J = 1, 149
   DO 2 K = J + 1, 50
   IF(SUBQ(K, 3).EQ.0) GO TO 2
   IF(SUBQ(J, 2), LE, SUBQ(K, 2)) GO TO 2
   DO 2 L = 1, 4
   M = SUBQ(J, L)
   SUBQ(J, L) = SUBQ(K, L)
   SUBQ(K, L) = M
2 CONTINUE

SORT LINE QUEUE
3 DO 1 I = 1, 5
   DO 3 J = 1, 5
   DO 3 K = J + 1, 50
   IF(LINEQ(I, K, 4).EQ.0) GO TO 3
   IF(LINEQ(I, J, 3), LE, LINEQ(I, K, 3)) GO TO 3
   DO 3 L = 1, 5
   M = LINEQ(I, J, L)
   LINEQ(I, J, L) = LINEQ(I, K, L)
   LINEQ(I, K, L) = M
3 CONTINUE
5 IF(IPRINT.GT.3) GO TO 4
   CALL SPRINT(SUBQ, WEEK, AFILE, REQWEEK)
   CALL LPRINT(LINEQ, WEEK, AFILE, REQWEEK)

SEGMENT FOR SUB-ASSEMBLIES
4 DO 18 J = 1, 50
   IF(WKCAP, LE. 0) GO TO 19
   IF(SUBQ(J, 3), EQ. 0) GO TO 18
   IF(SUBQ(J, 4), GT. 0) GO TO 18
   IPRD = SUBQ(J, 1)
   IF(SUBQ(J, 3), GT. WKCAP) GO TO 12
   IQTY = SUBQ(J, 3)
   GO TO 14
12 IQTY = WKCAP
14 WKCAP = WKCAP - IQTY
   SUBQ(J, 3) = SUBQ(J, 3) - IQTY
   IFILE(IPRD, 16) = IFILE(IPRD, 16) + IQTY
   IFILE(IPRD, 18) = IFILE(IPRD, 18) + IQTY
IF(IPRINT.GT.1)GO TO 18
WRITE(2,16)AFILE(IPROD,1),SUBQ(J,2),QTY
16 FORMAT(1HO,6X,ISUB=ASSEMBLY",2X,A8,2X,'DUE WEEK',I6,2X,
QTY,'16,2X,'MOVED TO STOCK")
18 CONTINUE
19 DO 180 MM=1,49
IF(SUBQ(MM,3).GT.0)GO TO 180
DO 170 M=MM+1,50
IF(SUBQ(M,3).GT.0)GO TO 175
170 CONTINUE
GO TO 10
175 DO 180 N=1,4
SUBQ(MM,N)=SUBQ(M,N)
SUBQ(M,N)=0
180 CONTINUE
10 DO 200 I=1,5
IF(IFILE(I,11).NE.1)GO TO 200
C SEGMENT FOR EQUIPMENTS
C IF(IPRINT.GT.1)GO TO 30
WRITE(2,550)AFILE(I,11),REQFILE(I,WEek)
550 FORMAT(1HO,6X,OFFLINE QUANTITY FOR",2X,A8,2X,'IS',I7)
30 IF(REQFILE(I,WEek).LE.0)GO TO 200
DO 120 J=1,50
IF(LINEQ(I,J,5).GT.0)GO TO 120
DO 110 J=1,54
QTY=LINEQ(I,J,4)
IF(LINEQ(I,J,4).EQ.0)GO TO 120
QTY=REQFILE(I,WEek)
110 CONTINUE
120 CONTINUE
10 DO 200 I=1,5
IF(IFILE(I,11).NE.1)GO TO 200
C SEGMENT FOR CUSTOMER ORDERS
C DO 40 K=1,50
IF(TEStQ(I,K,4).EQ.0)GO TO 50
40 CONTINUE
C IF(IPRINT.GT.5)GO TO 150
WRITE(2,45)AFILE(I,1)
45 FORMAT(1HO,6X,'EXCEEDED TEST QUEUE SIZE FOR PRODUCT NO.',2X,A8)
GO TO 150
50 JA=LINEQ(I,J,7)
JB=LINEQ(I,J,2)
JC=LINEQ(I,J,3)
DO 55 KEYNO=1,125
IF(OPFILE(KEYNO,JB,1,10).EQ.JA)GO TO 57
55 CONTINUE
GO TO 65
57 DO 60 M=1,5
IF(OPFILE(KEYNO,JB,M,3).EQ.JC)GO TO 80
60 CONTINUE
C IF(IPRINT.GT.5) GO TO 120
WRITE(2,70)JA,JB,JC,OPFILE(KEYNO,JB,M,3),REFWEEK
70 FORMAT(1HO,6X,'INCOMPATIBLE DATA IN OPFILE FOR ORDER NO.',I4,' Item 1M NO.',I4,'DUE WEEK',I4/6X,'ACTUAL WEEK',I4,6X,'REFWEEK',I14)
IPRINT=1
DO 75 IZ=1,10
75 DIAGFILE(IZ,2)=1
RUNTIME=REFWEEK+WEek+4
GO TO 120
80 DO 90 L=1,3
90 TESTQ(I;K;L)=LINEQ(I;J;L)
    TESTQ(I;K;4)=QTY
    TESTQ(I;K;5)=3
    IFILE(I,19)=IFILE(I,19)+QTY
    OPFILE(KeVNO;J,K;M,8)=OPFILE(KeVNO;J,K;M,8)+QTY
    OPFILE(KeVNO;J,K;M,7)=OPFILE(KeVNO;J,K;M,7)-QTY
    GO TO 110
100 IFILE(I,16)=IFILE(I,16)+QTY
110 LINEQ(I;J;4)=LINEQ(I;J,4)-QTY
    IFILE(I,18)=IFILE(I,18)+QTY
    REQFILE(I,week)=REQFILE(I,week)+QTY
    IF(IPRINT.GT.3)GO TO 115
    WRITE(2,530)(LINEQ(I;J,K),K=1,3)+QTY
530 FORMAT(1HO,6X,ORDER',I5,2X,ITEM',I5,2X,DUE',I5,2X,QUANTITY',I5,12X,Moving Off Line')
115 IF(REQFILE(I,week).LE.0)GO TO 150
120 CONTINUE
500 IF(IPRINT.GT.2)GO TO 150
    WRITE(2,510)AFILE(I,1)
510 FORMAT(1HO,6X,EXHAUSTED ORDERS IN W.I.P. FOR PRODUCT',2X,AB)
150 DO 160 MM=1,49
    IF(LINEQ(I,MM,4).GT.0)GO TO 160
    DO 153 M=(MM+1),50
    IF(LINEQ(I,M,4).GT.0)GO TO 155
153 CONTINUE
    GO TO 200
155 DO 158 N=1,5
    LINEQ(I,MM,N)=LINEQ(I,MM,N)
    LINEQ(I,MM,N)=0
158 CONTINUE
160 CONTINUE
200 CONTINUE
    IF(IPRINT.GT.2)GO TO 250
    CALL TQPRINT(TESTQ,week,AFILE,REFWEEK)
250 RETURN
END
Orders received history is maintained in file ORDHIST, and is used to prepare the orders received forecast in subroutine FORECAST. ORDHIST is organised into twelve monthly buckets per product as shown below:

As orders are generated and entered into the order book in subroutine LOAD, the current month bucket in ORDHIST for the appropriate product is augmented by the order quantity.

At the end of each month the file is reset in subroutine OHRESET by shifting each entry in ORDHIST by one period. Values for month (-11) are removed from the file and the current month is cleared down in preparation for new orders received data.
SUBROUTINE OHRESET (ORDHIST)

DO
L = 1,5

DO
J = 1,11

K = J + 1

ORDHIST (L,J) = ORDHIST (L,K)

ORDHIST (L,K) = 0

RETURN

END

Summarised Flow Chart - Subroutine Ohreset
Subroutine OPFCREATE is used to create the initial conditions, where these are related to the orders placed file (OPFILE) records.

Customer order information is read from the source file and creates an OPFILE record, which subsequently is used to create a corresponding order book (ORDBK) record for the appropriate product. The total load in file TOTAL is incremented by the order quantity and items "to be allocated ex-stock" will increment the allocated stock filed in IFILE. Items shown "in allocation" status will cause the component stock to be allocated by making use of subroutine STKALL, which will also create a record in the allocation queue ALLQ. Items shown in "on-line" status will cause a record to be created in file LINEQ and items in "test" status will create a record in TESTQ.

When the source data file has been exhausted, the work in progress quantity for each product is derived by summing the quantities in LINEQ and TESTQ, and the totals are entered in IFILE. The sub-assembly work in progress, contained in file SUBQ is also totalled for entry in IFILE. SUBQ is initialised in the main STOCKMODEL segment.
Summarised Flow Chart - Sub-routine Opfcreate
SUBROUTINE OPFCREATE(OPFILE, ORDBK, TOTAL, ALLQ, LINEQ, TESTQ, ORDCOUNT, REFWEEK, IFILE, SEED4, ICOUNT, ISHORT, IPRINT, AFILE, SUBQ)

REAL AFILE(25, 3)
INTEGER OPFILE(125, 575, 10), ORDBK(5, 52, 20, 5), ORDNO, ITEMNO, DUE, QTY,
TOTAL(5, 52), ALLQ(1250, 6), LINEQ(5, 50, 5), TESTQ(5, 50, 5), ORDCOUNT

2!PROD, REFWEEK, IFILE(25, 23), SUBQ(50, 4), ICOUNT(25), ISHORT(25)

IF(IPRINT, GT, 1) GO TO 5
WRITE(2, 3)
3 FORMAT(1HO, 6X, !* * * * OPFCREATE * * * * */)
5 ORDCOUNT = 0
DO 170 1 = 1, 125
READ(1, 10) NUMBER, IPS
10 FORMAT(215)

NUMBER = NUMBER OF ITEMS, NWKS = NUMBER OF WEEKS OF PHASING

IF(NUMBER, EQ, 0) GO TO 175
DO 170 J = 1, NUMBER
READ(1, 15) NWKS
15 FORMAT(15)
DO 170 K = 1, NWKS
READ(1, 20) (OPFILE(I, J, K, L), L = 1, 10)
20 FORMAT(1015)
ORDNO = OPFILE(I, J, K, 10)
IF(ORDNO, LE, ORDCOUNT) GO TO 25
ORDCOUNT = ORDNO
25 ITEMNO = J
QTY = OPFILE(I, J, K, 2) - OPFILE(I, J, K, 9)
DUE = OPFILE(I, J, K, 3)
PROD = OPFILE(I, J, K, 1)

FIND NEXT SPACE IN ORDER BOOK

DO 30 M = 1, 20
IF(ORDBK(PROD, DUE, M, 1), EQ, 0) GO TO 50
30 CONTINUE
IF(IPRINT, GT, 5) GO TO 200
WRITE(2, 40) PROD
40 FORMAT(1HO, 6X, !ORDER BOOK SIZE EXCEEDED FOR PRODUCT NUMBER', I6, 2X, '
DUE WEEK', I6)
GO TO 200
50 ORDBK(PROD, DUE, M, 1) = ORDNO
ORDBK(PROD, DUE, M, 2) = ITEMNO
ORDBK(PROD, DUE, M, 3) = QTY
ORDBK(PROD, DUE, M, 4) = 0
ORDBK(PROD, DUE, M, 5) = IPS
TOTAL(PROD, DUE) = TOTAL(PROD, DUE) + QTY
IF(OPFILE(I, J, K, 5), EQ, 0) GO TO 80

ALLOCATE COMPONENTS

IFILE(PROD, 17) = IFILE(PROD, 17) + OPFILE(I, J, K, 5)
80 IF(OPFILE(I, J, K, 6), EQ, 0) GO TO 90

MOVE ITEM INTO ALLOCATION QUEUE AND RESERVE COMPONENTS
NR=OPFILE(I,J,K,6)
IMAX=0
OUTPUT=0.0
CALL STKALL(IFILE, NR, PROD, ALLQ, ORDNO, ITEMNO, DUE, OUTPUT, ICOUNT
IF(ISHORT, IPRINT, IMAX, AFILE, &200)
90 IF(OPFILE(I,J,K,7).EQ.0) GO TO 130

C FIND NEXT SPACE IN LINE QUEUE
DO 100 L=1,50
IF(LINEQ(PROD,L,1).EQ.0) GO TO 120
100 CONTINUE
IF(IPRINT.GT.5) GO TO 200
WRITE(2,110) PROD
110 FORMAT(1HO,6X,1EXCEEDED LINE QUEUE FOR PRODUCT NUMBER1,16)
GO TO 200
120 LINEQ(PROD,L,1)=ORDNO
LINEQ(PROD,L,2)=ITEMNO
LINEQ(PROD,L,3)=DUE
LINEQ(PROD,L,4)=OPFILE(I,J,K,7)
LINEQ(PROD,L,5)=0
130 IF(OPFILE(I,J,K,8).EQ.0) GO TO 170

C FIND NEXT SPACE IN TEST QUEUE
DO 140 L=1,50
IF(TESTQ(PROD,L,1).EQ.0) GO TO 160
140 CONTINUE
IF(IPRINT.GT.5) GO TO 200
WRITE(2,150) PROD
150 FORMAT(1HO,6X,1EXCEEDED TEST QUEUE FOR PRODUCT NUMBER1,16)
GO TO 200
160 TESTQ(PROD,L,1)=ORDNO
TESTQ(PROD,L,2)=ITEMNO
TESTQ(PROD,L,3)=DUE
TESTQ(PROD,L,4)=OPFILE(I,J,K,8)
TESTQ(PROD,L,5)=0
170 CONTINUE

C LOAD WIP RECORDS
175 DO 190 I=1,5
IF(IFILE(I,11).NE.1) GO TO 190
IFILE(I,18)=0
DO 180 J=1,50
IFILE(I,J)=IFILE(I,18)+LINEQ(I,J,4)
180 CONTINUE
IFILE(I,19)=0
DO 190 J=1,50
IFILE(I,J)=IFILE(I,19)+TESTQ(I,J,4)
190 CONTINUE
DO 195 I=1,20
IF(IFILE(I,11).EQ.2) GO TO 103
IF(IFILE(I,11).NE.3) GO TO 195
193 DO 195 J=1,50
IF(SUBQ(J,1).NE.1) GO TO 195
IFILE(I,J)=IFILE(I,18)+SUBQ(J,3)
195 CONTINUE
200 RETURN
END
Subroutine OPFPRT is used to display the contents of file OPFILE for diagnostic purposes. Orders are displayed in the sequence held in the file, which is quasi-random. Redundant entries which have been cleared down are not displayed.

Data is presented as week due within item number within order number, each line specifying the product type, ordered quantity, week due and status information.

A record count is displayed each time the subroutine is activated, to allow the modeller to remain within the file volume constraints of OPFILE.

Summarised Flow Chart - Subroutine Opfpert
SUBROUTINE OPFPRT(OPFILE;AFILE)
REAL AFILE(25,3)
INTEGER OPFILE(125,5,75,10)
WRITE(2,200)
200 FORMAT(1HO,6X,ORDER FILE DETAIL,50X,STATUS//6X,ORDER NUMBER 1 ITEM PRODUCT QUANTITY DUE DATE OPEN EX-STK ALN ON-LINE TEST DESP./')
ISTATUS=0
DO 110 I=1,125
IF(OPFILE(I,1,1,10).EQ.0)GO TO 110
WRITE(2,210)OPFILE(I,1,10)
210 FORMAT(1H1,6X,FILE DETAILI,50X,ISTATUS'!!6X,IORDER NUMnBER ITEM PRODUCT QUANTITY DUE DATE OPEN EX-STK ALN ON-LINE TEST DESP./')
ISTATUS=ISTATUS+1
DO 110 J=1,5
DO 110 K=1,5
IF(OPFILE(I,J,K,1).EQ.0)GO TO 110
WRITE(2,205)J
205 FORMAT(1H1,22X,ISTATUS')
N=OPFILE(I,J,K,1)
WRITE(2,220)AFILE(N),OPFILE(I,J,K,L),L=2,9
220 FORMAT(1H1,31X,A8,S-219;616/) 110 CONTINUE
WRITE(2,230)ISTATUS
230 FORMAT(1H1,6X,RECORDER COUNT =',16) RETURN
END

ORDER-FILE DETAIL
ORDER NUMBER ITEM PRODUCT QUANTITY DUE DATE OPEN EX-STK ALN ON-LINE TEST DESP.
24 1 MF6AM 03 4 26 0 4 0 0 0 0
1 MF6AM 03 21 27 0 21 0 0 0 0
2 MF6AM 02 9 27 1 0 0 0 0 0
2 MF6AM 02 7 28 7 0 0 0 0 0
3 MF6AM 02 33 28 33 0 0 0 0 0
3 MF6AM 02 17 29 17 0 0 0 0 0
2 1 MF6AM 01 20 13 0 0 0 0 0 0
1 MF6AM 01 20 14 0 0 0 0 0 0
1 MF6AM 01 20 15 0 0 0 0 0 0
1 MF6AM 01 20 16 0 0 0 0 0 0
1 MF6AM 01 20 17 0 0 0 0 0 0
1 MF6AM 02 2 25 0 0 0 0 0 0
18 1 MF6AM 01 20 13 0 0 0 0 0 0
1 MF6AM 01 20 14 0 0 0 0 0 0
1 MF6AM 01 20 15 0 0 0 0 0 0
1 MF6AM 01 20 16 0 0 0 0 0 0
1 MF6AM 01 20 17 0 0 0 0 0 0
27 1 MF6AM 01 13 29 13 0 0 0 0 0
1 MF6AM 01 12 30 12 0 0 0 0 0
2 MF6AM 02 23 29 23 0 0 0 0 0
2 MF6AM 02 27 30 27 0 0 0 0 0
3 MF6AM 01 10 30 10 0 0 0 0 0
4 MF6AM 01 5 30 5 0 0 0 0 0
5 MF6AM 01 1 30 1 0 0 0 0 0
5 1 MF6AM 03 25 17 0 0 0 0 0 25
2 MF6AM 01 50 14 0 0 0 0 0 0
2 MF6AM 01 50 15 0 0 0 0 0 0
3 MF6AM 01 50 16 0 0 0 0 0 0
3 MF6AM 01 25 17 0 0 0 0 0 0
4 MF6AM 02 10 17 0 0 0 0 0 0
5 AC15PU 20 16 0 0 0 0 0 0
28 1 AC15PU 25 30 25 0 0 0 0 0
1 AC15PU 25 31 25 0 0 0 0 0
2 MF6AM 01 10 30 10 0 0 0 0 0
3 MF6AM 01 1 30 1 0 0 0 0 0

Sample Orders Placed File
Subroutine OROBKPRT is used to display the content of the order book, ORDBK, for each product as a diagnostic tool.

The print format is arranged in product sequence and displays as total values the overdue load and the total order load. For due orders, the order number, item number and quantity outstanding are displayed, with an indicator signifying part shipment or no part shipment allowed.

The print format will accommodate up to 20 weeks of due orders. If live orders exist beyond this horizon a further 20 weeks horizon is displayed. Orders beyond 40 weeks are not considered in OROBKPRT.

Each week in ORDBK can accommodate a maximum of fifteen entries. OROBKPRT will suppress zero records, thus reducing the volume of printed output. A counter is contained within the subroutine to register the maximum record count, indicating to the modeller the proportion of file ORDBK utilised during each simulation run.
SUBROUTINE  
ORDBKPR

SELECT  
NEXT  
PRODUCT

CALCULATE  
TOTAL  
LOAD

PRINT  
HEADER

* REPEAT

PRINT  
TOTAL

PRINT  
ORDER  
NO.

PRINT  
ITEM  
NO.

PRINT  
QUANTITY

PRINT  
PS/NPS

INCREMENT  
RECORD  
COUNT

* REPEAT

PRINT  
RECORD  
COUNT

RETURN

* REPEATED FOR
1. Weeks 1 through 20
2. Weeks 21 through 40

Summarised Flow Chart - Subroutine Ordbkprt
SUBROUTINE ORDBKPRT(ORDBK, AFILE, IFILE, TOTAL, REFWEEK)
REAL AFILE(25,3)
INTEGER ORDBK(5,52,20;5), IFILE(25,23), TOTAL(5,52), REFWEEK
MAX2 = 0
DO 100 I = 1,5
IF(IFILE(I,11).NE.;1) GO TO 100
WRITE(2,10) AFILE(I,1)
10 FORMAT(1HO,6X,ORDERT BOOK DETAIL FOR PRODUCT NUMBER,2X,A8)
INDEX = 0
ISUM = 0
DO 20 J = 1,52
DO 20 K = 1,20
IF(ORDBK(I,J,K;4;).EQ.,0) GO TO 20
ISUM = ISUM + ORDBK(I,J,K;3)
20 CONTINUE
L1 = 1
L2 = 20
M = REFWEEK + 1
N = REFWEEK + 20
30 WRITE(2,30) (J,J=M:N)
30 FORMAT(1HO,6X,ITEM,10X,2X,0/D,2X,2014)
WRITE(2,33) TOTAL(I;J;J=L1;L2)
33 FORMAT(1HO,6X,ORDER BOOK DETAIL FOR PRODUCT NUMBER,2X,A8)
INDEX = D
ISUt1 = 0
DO 40 J = 1,52
DO 40 K = 1,20
IF(ORDBK(I,J,K;4;).EQ.,0) GO TO 40
ISUM = ISUM + ORDBK(I,J,K;3)
40 CONTINUE
L1 = 1
L2 = 20
N = REFWEEK + 20
50 WRITE(2,50) (J,J=L1:L2)
50 FORMAT(1HO,6X,WRITE,10X,2014)
WRITE(2,53) TOTAL(I;J;J=L1;L2)
53 FORMAT(1HO,6X,ORDER BOOK DETAIL FOR PRODUCT NUMBER,2X,A8)
INDEX = 0
ISUM = 0
DO 60 K = 1,20
DO 60 J = L1, L2
IF(ORDBK(I,J,K;4;).EQ.,1) GO TO 60
IF(ORDBK(I,J,K;3;).GT.,O) GO TO 60.
60 CONTINUE
GO TO 90
57 WRITE(2,60) (ORDBK(I,J,K;1;),J=L1:L2)
60 FORMAT(1HO,6X,ORDER BOOK DETAIL FOR PRODUCT NUMBER,2X,A8)
WRITE(2,70) TOTAL(I;J;J=L1;L2)
70 FORMAT(1HO,6X,ITEM,10X,2X,0/D,2X,2014)
WRITE(2,70) TOTAL(I;J;J=L1;L2)
80 FORMAT(1HO,6X,ORDER BOOK DETAIL FOR PRODUCT NUMBER,2X,A8)
INDEX = D
ISUt1 = 0
DO 85 K = 1,20
DO 85 J = L1, L2
IF(ORDBK(I,J,K;4;).EQ.,1) GO TO 85
IF(ORDBK(I,J,K;3;).EQ.,O) GO TO 85
85 CONTINUE
GO TO 90
INDEX = 0
GO TO 30
85 CONTINUE
GO TO 90
INDEX = 0
GO TO 30
90 CONTINUE
IF(MAX1.GT.MAX2) GO TO 92
MAX2 = MAX1
92 IF(INDEX.EQ.1) GOTO 100
DO 95 J = 21,40
DO 95 K = 1,20
IF(ORDBK(I,J,K;4;).EQ.,1) GO TO 95
IF(ORDBK(I,J,K;3;).EQ.,O) GO TO 95
M = REFWEEK + 21
N = REFWEEK + 40
ISUM = 0
L1 = 21
L2 = 40
INDEX = 1
GO TO 30
95 CONTINUE
100 CONTINUE
WRITE(2,110) MAX2
110 FORMAT(1HO,6X,MAXIMUM RECORD COUNT =', I6)
RETURN
END
## Sample Product Order Book

<table>
<thead>
<tr>
<th>WEEK</th>
<th>Q/D</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
<th>31</th>
<th>32</th>
<th>33</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOAD</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>25</td>
<td>0</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>ORD, NO.</td>
<td></td>
<td>0</td>
<td>5</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>0</td>
<td>31</td>
<td>35</td>
<td>18</td>
<td>24</td>
<td>16</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>27</td>
<td>30</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>ITEM</td>
<td></td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>QTY</td>
<td></td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>0</td>
<td>10</td>
<td>7</td>
<td>2</td>
<td>34</td>
<td>1</td>
<td>17</td>
<td>17</td>
<td>27</td>
<td>14</td>
<td>11</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>T=PS:2=NPS</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>ORD, NO.</td>
<td></td>
<td>9</td>
<td>0</td>
<td>4</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>35</td>
<td>0</td>
<td>35</td>
<td>0</td>
<td>16</td>
<td>24</td>
<td>27</td>
<td>29</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>ITEM</td>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QTY</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>24</td>
<td>0</td>
<td>32</td>
<td>0</td>
<td>35</td>
<td>23</td>
<td>23</td>
<td>26</td>
<td>29</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T=PS:2=NPS</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>ORD, NO.</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ITEM</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QTY</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T=PS:2=NPS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**ORDER BOOK DETAIL FOR PRODUCT NUMBER MF6AM 02**
Subroutine ORDERS simulates the arrival of customer orders into the order processing pipeline. The structure and profile of orders generated is a reasonable approximation to the pattern observed in the real world.

The subroutine makes use of five sampling profiles, determining:

- the number of orders to be generated in each period (simulated week)
- the number of order lines
- the product applicable to each order line
- the quantity of each product
- whether part shipment is permissible

The orders pipeline file, PIPELINE, is cleared down to accept new orders and the number of new orders is determined by sampling from a normal distribution in subroutine NORMAL. The mean and coefficient of variation of the distribution may be modified by the modeller, but it should be noted that a mean in excess of 5.0 may cause file volumes to be exceeded.

For each new order, the number of items is established by sampling from the frequency histogram contained in TABLE 5, using subroutine HSAMPLE 1. The "part shipment" or "no part shipment" condition is determined by sampling from TABLE 8 in subroutine HSAMPLE 2.

Two further frequency histograms are sampled for each order item line; the product to be selected and the quantity required. Both use HSAMPLE 1 and the parameters from TABLE 6 and TABLE 7 respectively. The quantity for each product is extended by the unit cost contained in IFILE to arrive at an approximate sales value for later analysis if required.

The resultant customer order profile is entered as a new record in the file PIPELINE for subsequent order loading.
Summarised Flow Chart - Subroutine Orders
SUBROUTINE ORDERS(TABLE5, TABLE6, TABLE7, ORDCOUNT, PIPELINE, ISEE4, ISEE5, ISEE6, ISEE7, AVGE4, CVARN4, IFILE, TABLE8, ISEE8, IPRINT)
INTEGER PIPELINE(20,6,4), TABLE5(7,2), TABLE6(7,2), TABLE7(7,2), ORDCOUNT, PROD, QTY, IFILE(25,23), TABLE8(2,2)

CLEAR PIPELINE

DO 5 I=1,20
   DO 5 J=1,6
      DO 5 K=1,4
         PIPELINE(I,J,K)=0
      5 CONTINUE

SET NUMBER OF ORDERS

CALL NORMAL(AVGE4, X, ISEE4, CVARN4)
   IF(X.GE.0.0) GO TO 10
   X=0.0
10 IF(X.LE.20.0) GO TO 20
   X=20.0
20 NUMBER=NINT(X)
   IF(IPRINT.GT.1) GO TO 30
   WRITE(2,500) ISEE04, X, NUMBER
500 FORMAT(1HD,6X,110;6X,F0:5,6X,I10)
30 DO 40 K=1, NUMBER
   ORDCOUNT=ORDCOUNT+1

DEFINE ORDER CONTENT ... NUMBER OF ITEMS

CALL HSAMPLE 1(ISEE4, ITEMCOUNT, TABLE5)
   PIPELINE(K,1,1)=ORDCOUNT

ESTABLISH WHETHER PART SHIP OR NOT

CALL HSAMPLE 2(ISEE8, NO, TABLE8)
   PIPELINE(K,1,2)=NO
   PIPELINE(K,1,3)=ITEMCOUNT
   DO 40 I=1, ITEMCOUNT
      L=I+1

DEFINE EQUIPMENT AND QUANTITY

CALL HSAMPLE 1(ISEE6, PROD, TABLE6)
   CALL HSAMPLE 1(ISEE7, QTY, TABLE7)
   PIPELINE(K,L,1)=I
   PIPELINE(K,L,2)=PROD
   PIPELINE(K,L,3)=QTY
   PIPELINE(K,L,4)=1.7*(IFILE(PROD,13)*1.2+IFILE(PROD,14)*3.5)*QTY
40 CONTINUE
RETURN
END
Subroutine PLNPRT is used to display the contents of the orders to be loaded pipeline, PIPELINE, for diagnostic use.

PLNPRT will display all new orders generated by subroutine ORDERS prior to loading in subroutine LOAD. Only active records containing new order numbers are displayed.

For each order number, PLNPRT will indicate each comprising item number and details of the products and quantities selected. The value is also displayed if further analysis is required by the modeller. Each order entry is accompanied by a statement signifying whether or not part shipment is permissible.
SUBROUTINE PLNPRINT(PIPELINE, AFILEx)
REAL AFILEx(25,3)
INTEGER PIPELINE(20,6,4)
WRITE(2,10)
10 FORMAT(1H0,20X, 'CONTENTS OF PIPELINE\n',//6X,'ORDER NO',1,6X,'ITEM NO',1,3X,'EQUIPMENT',3X,'QUANTITY',3X,'VALUE\n')
N=1
20 WRITE(2,30)PIPELINE(N,1,1)
30 FORMAT(1H0,9X,13)
DO 40 I=1,PIPELINE(N,1,3)
L=I+1
K=PIPELINE(N,L,2)
40 WRITE(2,50)PIPELINE(N,L,1),AFILE(K,1),(PIPELINE(N,L,J),J=3,4)
ICHECK=PIPELINE(N,1,2)
GO TO(60,80)ICHECK
60 WRITE(2,70)
70 FORMAT(1H0,6X,'PART SHIPMENT ALLOWED/')
GO TO 95
80 WRITE(2,90)
90 FORMAT(1H0,6X,'NO PART SHIPMENT ALLOWED/')
95 N=N+1
IF(PIPELINE(N,1,1).NE.0)GO TO 20
100 RETURN
END

CONTENTS OF PIPELINE

<table>
<thead>
<tr>
<th>ORDER NO.</th>
<th>ITEM NO.</th>
<th>EQUIPMENT</th>
<th>QUANTITY</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1</td>
<td>HF6AM 02</td>
<td>1</td>
<td>179</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>HF6AM 02</td>
<td>5</td>
<td>897</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>HF6AM 01</td>
<td>10</td>
<td>1795</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>AC15PU</td>
<td>25</td>
<td>2018</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>AC15PU</td>
<td>2</td>
<td>161</td>
</tr>
</tbody>
</table>

NO PART SHIPMENT ALLOWED

<table>
<thead>
<tr>
<th>ORDER NO.</th>
<th>ITEM NO.</th>
<th>EQUIPMENT</th>
<th>QUANTITY</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>1</td>
<td>HF6AM 01</td>
<td>5</td>
<td>897</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>HF6AM 01</td>
<td>10</td>
<td>1795</td>
</tr>
</tbody>
</table>

PART SHIPMENT ALLOWED

Sample Pipeline Contents Report
SUBROUTINE PNMFCREATE

The subroutine PNMFCREATE is the means of loading data into the master files. The master data is held within two files; AFILE, which is a real array and IFILE, an integer array.

The data which is read into the master files is of three types:

a) Fixed data, including structure relationships, cost data, operational parameters and descriptive information.

b) Semi-variable parameters, which may be defined prior to each simulation experiment (e.g., Lead time, buffer level).

c) Variable data, including stock and work-in-progress status, input as initial conditions.

The file detail is defined more fully in the file descriptions for AFILE and IFILE.
SUBROUTINE PNMFCREATE(AFILE, IFILE)
REAL AFILE(25,3)
INTEGER IFILE(25,23)
DATA BLANK /1/
DO 30 I=1,25
DO 10 J=1,23
10 IFILE(I,J)=0
READ(1,20)(AFILE(I,J),J=1,3),(IFILE(I,J),J=1,16)
20 FORMAT(3A8,1512,I4)
CALL COMPIL(AFILE(I,1),BLANK,ITEST)
GO TO (40,30),ITEST
30 CONTINUE
40 RETURN
END
The subroutine PNMFPRINT is used to format and present the detail contained within the master data files AFILE and IFILE.

The master data will usually be printed at the commencement of a simulation experiment to indicate the initial conditions pertaining to each master item.

**Summarised Flow Chart - Subroutine PNMFPRINT**

```
SUBROUTINE PNMFPRTN(AFILE, IFILE)
REAL AFILE(25,3)
INTEGER IFILE(25,23)
WRITE(2,10)
10 FORMAT(1H1,7X,'PART NO, OESCRIPTION AD QY AD QY AD QY AD QY AD Q
  Y LE LD MT LB IS PHYS ALLD LINE TEST DU EBQ CY AFS!1/64X,'TM CT CT
  2CD STCK STCK WIP WIP WK TM MAX')
DO 20 I=1,25
20 WRITE(2,30)I,(AFILE(I,J),J=1,3),,(IFILE(I,J),J=1,23)
30 FORMAT(1H0,1X,12,2X,3A8,15137415;13,14,14,14)
RETURN
END
```
| PART NO. | DESCRIPTION | AD | QY | AD | QY | AD | QY | LE | LD | MT | LB | IS | PHYS | ALLD | LINE | TEST | BU | EQB | CY | AFS | TM | CT | CT | CO | STCK | STCK | WIP | WIP | UK | TM | MAX |
|----------|-------------|----|----|----|----|----|----|----|----|----|----|----|------|------|------|------|-----|----|----|----|----|------|------|-----|-----|----|----|-----|
| 1 | MF6AM 01RADIO TELEPHONE | 5 | 1 | 6 | 1 | 19 | 1 | 11 | 1 | 12 | 1 | 1 | 2 | 53 | 12 | 1 | 1 | 48 | 169 | 260 | 164 | 4 | 0 | 0 | 334 |
| 2 | MF6AM 02RADIO TELEPHONE | 5 | 1 | 7 | 1 | 19 | 1 | 11 | 1 | 12 | 1 | 1 | 2 | 53 | 12 | 1 | 30 | 147 | 155 | 81 | 4 | 0 | 0 | 154 |
| 3 | MF6AM 03RADIO TELEPHONE | 5 | 1 | 8 | 1 | 19 | 1 | 11 | 1 | 12 | 1 | 1 | 2 | 53 | 12 | 1 | 63 | 58 | 55 | 43 | 4 | 0 | 0 | 145 |
| 4 | AC15PU POWER UNIT | 23 | 1 | 9 | 1 | 15 | 1 | 17 | 1 | 14 | 1 | 1 | 2 | 25 | 5 | 1 | 0 | 88 | 26 | 4 | 0 | 0 | 114 |
| 5 | AT12345 L.F. ASSEMBLY | 11 | 2 | 14 | 1 | 21 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 25 | 5 | 2 | 786 | 262 | 828 | 0 | 4 | 160 | 1 | 0 |
| 6 | AT12347 R.F. ASSEMBLY | 12 | 3 | 18 | 1 | 22 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 25 | 4 | 2 | 732 | 136 | 350 | 0 | 4 | 185 | 2 | 0 |
| 7 | AT12801 R.F. ASSEMBLY | 13 | 4 | 15 | 1 | 22 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 25 | 4 | 2 | 67 | 89 | 136 | 0 | 4 | 100 | 2 | 0 |
| 8 | AT12802 R.F. ASSEMBLY | 14 | 5 | 16 | 1 | 22 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 25 | 4 | 2 | 99 | 46 | 112 | 0 | 4 | 92 | 3 | 0 |
| 9 | AT27096 P.C.B. ASSY. | 15 | 6 | 17 | 1 | 10 | 1 | 0 | 0 | 0 | 0 | 2 | 4 | 12 | 2 | 2 | 153 | 0 | 242 | 0 | 4 | 122 | 4 | 0 |
| 10 | AT22000 P.C.R. ASSY. S1 | 16 | 7 | 17 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 4 | 9 | 1 | 1 | 0 | 0 | 0 | 0 | 130 | 7 | 0 |
| 11 | RU49873 SCREW 10NM M2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 10 | 1 | 0 | 2 | 6444 | 262 | 0 | 0 | 8 | 0 | 0 |
| 12 | RU10034 NUT M2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 10 | 1 | 0 | 2 | 1224 | 262 | 0 | 0 | 8 | 0 | 0 |
| 13 | BT49863 MAIN FRAME | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 20 | 18 | 0 | 2 | 1850 | 192 | 0 | 0 | 8 | 0 | 0 |
| 14 | PP42906 CAPACITOR 100P | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 15 | 1 | 0 | 2 | 2404 | 0 | 0 | 8 | 0 | 0 |
| 15 | PN50006 RESISTOR 110 OHM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 12 | 1 | 0 | 2 | 612 | 100 | 0 | 0 | 8 | 0 | 0 |
| 16 | PN10638 RESISTOR 5K OHM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 12 | 1 | 0 | 2 | 103 | 92 | 0 | 0 | 8 | 0 | 0 |
| 17 | PN10639 RESISTOR 10K OHM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 12 | 1 | 0 | 2 | 1246 | 0 | 0 | 8 | 0 | 0 |
| 18 | PN56043 RESISTOR 22K OHM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 12 | 1 | 0 | 2 | 1448 | 0 | 0 | 8 | 0 | 0 |
| 19 | PN69746 RESISTOR 100 OHM | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 12 | 1 | 0 | 2 | 1334 | 262 | 0 | 0 | 8 | 0 | 0 |
| 20 | ET12345 BOARD BLANK | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 16 | 6 | 0 | 2 | 418 | 0 | 0 | 8 | 0 | 0 |
| 21 | FU10000 INTEGRATED CCC. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 18 | 22 | 0 | 2 | 1696 | 0 | 0 | 8 | 0 | 0 |
| 22 | FV25000 POWER TRANSISTOR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 16 | 3 | 0 | 2 | 3410 | 384 | 0 | 0 | 8 | 0 | 0 |
| 23 | FS66000 SOCKET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 14 | 10 | 0 | 2 | 252 | 0 | 0 | 8 | 0 | 0 |
The subroutine QPLAN simulates the derivation of a production plan from the orders received forecast, known order book and stock status.

The programme is recalculated at quarterly intervals and is stated in 13 week periods. The programme is derived by reference to a number of pre-defined parameters, specifically:

a) the maximum authorised stock, or the level of finished equipment that is permitted to be held in stock in unallocated (or "free") status. The level of stock is re-assessed, within the Company, each 6 months, therefore, the model is expected to reflect this review within the logic. It should be noted that contention between the maximum authorised stock and other parameters may cause a plan which is temporarily in excess of the authorised level.

The opening stock is derived from the PNMF record for the product.

b) work-in-progress, which is considered to be the work-in-progress prior to the off-line point (thus excluding any test work-in-progress). The level of planned work-in-progress is derived from the manufacturing lead time as stated on the PNMF. The logic within the model will attempt to correct any difference between the actual opening work-in-progress and the nominal level and thereafter maintain the nominal level as defined by the lead time.

c) recommended running rate, or the weekly planned capacity of the product. The rates will normally reflect the resource in labour and plant to fully utilise a working week, although, for simplification, an assumption has been made in the model as an approximation to reality.
It is further assumed that:

(i) the rate cannot be changed within a period;
(ii) any rate change will be immediate (a step function);
(iii) all periods are a full 13 weeks with no adjustment for actual working days due to holidays.

d) minimum order book, which is the least number of orders that can be supported in the order book at a given capacity level. Typically, due to crystal lead times, this tends to be 6 weeks on a frequency conscious product and 4 weeks on a non-frequency conscious product.

e) nominal order book, or the number of weeks worth of orders (on average) that is set by policy to maintain a reasonable working backlog of work. Since there is no feedback from the final aggregate plan for all products which reflects variances against budgetted output or capacity, this parameter is often pre-defined in practice to offer some degree of control over the level of the plan in response to a variable orders received forecast.

The variables which are used in the calculation include:

a) opening work-in-progress as defined above, from the PNMF;

b) opening stock as defined above, also from the PNMF;

c) previous material plan, which is carried forward from the previous quarter's material plan for the product;

d) current order load, or the phased order book for the product adjusted in time to reflect the demand at the off-line point. (The plan makes no allowance for abnormal test work-in-progress). The "brought forward" figure represents delinquent performance, or overdue products (at the off-line point).
Using the relationship discussed in the main report narrative, namely:

- Orders carried forward = orders brought forward + orders received forecast - production to be allocated
- Closing stock = opening stock + off-line plan - production to be allocated
- Closing work-in-progress = opening work-in-progress + material plan - off-line plan.

The forward projection may be derived for the three reference plans:

- production to be allocated, which is used for order loading
- off-line plan, the plan against which main equipment shop schedules are geared
- material plan, the input to the purchasing and material scheduling routines.

The logic in developing the plan has been kept as simple as possible without destroying, as far as possible, the reality of the simulation. Thus, no facility has been included to regress to a previous period where the line running rate is changed, to avoid frequent rate changes. In practice this situation does not arise because the plan is derived manually and the planner is able to view the complete planning horizon as the decisions are made. Further, simple rules have been incorporated to reflect critical situations, for example, a sudden change in forecast which results in a temporary overstock or excess work-in-progress.

In the real world the planner may use some judgement in terms of a trade-off between stock of finished equipments, sub-assemblies or components.

The module QPLAN makes use of a secondary module LIMIT, which defines the constraints within which any changes to a plan may be made.
CALCULATE STOCK

STOCK INCREASING?

OFF-LINE PLAN STOCK INCREASE

NEW STOCK = OLD STOCK

CALCULATE PROD., TO BE ALLOCATED

CALCULATE NEW ORDER LOAD

NEGATIVE ORDER LOAD

INCREASE STOCK BY ORDER LOAD SHORTFALL

ORDER LOAD PROD., TO BE ALLOCATED

FLAG = 0?

OVERLOAD STOCK

REDUCE STOCK BY OVERLOAD

STOCK = 0

CALCULATE PROD., TO BE ALLOCATED

Cont. over.
CALCULATE WIP AND OFF-LINE PLAN

TEST FOR NEXT PERIOD > 0

CURRENT REQUIREMENTS > CURRENT AVAILABLE OFF-LINE PLAN + STOCK + WIP

NEW WIP = OLD WIP - REQUIREMENTS

NEW WIP = 0

NEW ORDER LOAD = REQUIREMENTS - OLD WIP

STOCK = OLD STOCK - NEW ORDER LOAD

IF STOCK > 0

STOCK = 0

CALCULATE PROD. TO BE ALLOCATED

CALCULATE OFF-LINE PLAN

CALCULATE MATERIAL PLAN

SET RUNNING RATE

CALCULATE NEW ORDER LOAD

Cont. over.
Summarised Flow Chart - Subroutine Qplan
SUBROUTINE QPLAN(ORF, COL, FES, WIP, IONP, R, AFSMAX, OCFMIN, OCFNOM, ONP, P1A, OCF, OFP, RR*, IFILE, REQFILE, IPRINT, MODE, EQUUBFR)

INTEGER ORF(10,7), COL(10,7), FES(10,7), WIP(10,7), IONP(10,7), R(10,10), AFSMAX(10), OCFMIN, OCFNOM, ONP(10,7), PA(10,10), FLAG, OCF(10,7), OFP(210,7), RR(10,11), ONPMAX, ONPMIN, IFILE(25,23), REQFILE(25,52), CUMDIF, EQUUBFR

RESET REQUIREMENTS FILES

DO 5 I=1,25
DO 5 J=1,52
5 REQFILE(I,J)=0
DO 200 L=1,10
IF(IFILE(L,11).NE.1) GO TO 200

SET START CONDITIONS

CUMDIF=0
DO 10 I=1,7
10 IONP(L,1)=IONP(L,1)
FES(L,1)=IFILE(L,16)
WIP(L,1)=IFILE(L,18)
OFP(L,1)=IONP(L,1)
IF(MODE.NE.3) GO TO 12
IFILE(L,23)=0
GO TO 19

12 K=0
DO 15 J=2,5
15 K=K+ORF(L,J)
IFILE(L,23)=K*EQUUBFR/52;0

SET FIRST QUARTER LOAD AND CALCULATE OPENING ORDERS

19 AFSMAX(L)=IFILE(L,23)
COL(L,2)=COL(L,2)*COL(L,1)
OCF(L,1)=0
DO 20 I=2,7
20 OCF(L,1)=OCF(L,1)*COL(L,1)

RESET QUARTER COUNTER

N=1
30 N=N+1
NN=N+1
NP=N+1
IF(NP.LE.7) GO TO 40
NP=7

SET PLAN CHANGE CONSTRAINTS

CALL LIMIT(OFP,L,NN);N, &170;IONP(L,N), ONPMAX, ONPMIN, CUMDIF)
IF(IPRINT.GT.1) GO TO 55
WRITE(2,50)L,N,ONPMAX,ONPMIN

50 FORMAT(1HO,6X,'PRODUCT = ',15,6X,'PERIOD = ',15,6X,'ONPMAX = ',15,6X,'ONPMIN = ',15)
55 FLAG=0
CALCULATE INITIAL OFF-LINE PLAN

J = 0
60 J = J + 1
   IF(J, LE: 10) GO TO 70
   J = J - 1
   GO TO 101
70 OFP(L, N) = R(ΔJ) * 13.0

CHECK FOR ZERO RUNNING RATE

IF(R(L, J), GT: 0) GO TO 80
OFP(L, N) = WIP(L; NN)
WIP(L, N) = 0
ONP(L, N) = 0
GO TO 90

CALCULATE NOMINAL WIP AND ON-LINE PLAN

80 WIP(L, N) = IFILE(L, 12) * R(ΔJ)
ONP(L, N) = OFP(L; NN) * WIP(L, NN) + WIP(L, N)

CHECK AGAINST MATERIAL PLAN CONSTRAINTS

90 IF(ONP(L, N), LE: ONPMAX) GO TO 100
FLAG = 1
   IF(J, EQ: 1) GO TO 100
   J = J - 1
   GO TO 70

FLAG IS SET TO 1 WHEN MAXIMUM MATERIAL PLAN IS REACHED

100 IF(ONP(L, N), GE: ONPMIN) GO TO 101
   IF(FLAG, EQ: 0) GO TO 60
   ONP(L, N) = ONPMIN
   IF(WIP(L; NN), GT: WIP(L, N)) GO TO 102
   J = J - 1
102 WIP(L, N) = IFILE(L, 12) * R(ΔJ)
OFP(L, N) = ONP(L; NN) + WIP(L; NN) + WIP(L, N)

CHECK FOR ZERO FORWARD REQUIREMENT

101 IF(ORF(L, N), GT: 0) GO TO 110
   IF(OCP(L, NN) + ORF(L; NN), GT: (RR(L; NN) * 13 + FES(L; NN) + WIP(L; NN))) GO TO 107
   WIP(L, N) = WIP(L; NN) - OCP(L; NN) - ORF(L, N)
   IF(WIP(L, N), LE: 0) GO TO 103
   FES(L, N) = FES(L; NN)
   GO TO 105
103 WIP(L, N) = 0
OCP(L, N) = OCP(L; NN) + ORF(L; NN) * WIP(L; NN)
FES(L, N) = FES(L; NN) - OCP(L, N)
   IF(FES(L, N), GT: 0) GO TO 105
   FES(L, N) = 0
105 PA(L, N) = OCP(L; NN) + ORF(L; NN)
OFP(L, N) = PA(L, N) + FES(L, N) + FES(L; NN)
GO TO 109
107 OFP(L, N) = RR(L; NN) * 13
FES(L, N) = FES(L; NN)
PA(L, N) = OFP(L, N)
WIP(L, N) = RR(L; NN) * IFILE(L, 12)
GO TO 162

CALCULATE STOCK AND PRODUCTION TO BE ALLOCATED

110 FES(L,N) = IFILE(L,23)/2.0

IS STOCK INCREASING

IF(FES(L,N) .LE. FES(L,NN)) GO TO 120
IF(OFP(L,N) .GT. FES(L,N) - FES(L,NN)) GO TO 120
FES(L,N) = FES(L,NN)

120 PAC(L,N) = OFP(L,N) - FES(L,N) + FES(L,NN)
OCF(L,N) = OCF(L,NN) + ORF(L,N) * PAC(L,N)
IF(OCF(L,N)) 124, 126, 126

124 FES(L,N) = FES(L,N) + OCF(L,N)
GO TO 120

CHECK FOR OVERLOAD

126 IF(COL(L,N) .LE. PAC(L,N)) GO TO 140
IF(FLAG.EQ.0) GO TO 60
IF((COL(L,N) - PA(L,N)) .GT. FES(L,N)) GO TO 130
FES(L,N) = FES(L,NN) + COL(L,N) + PAC(L,N)
GO TO 120

130 FES(L,N) = 0
PA(L,N) = OFP(L,N) + FES(L,NN)

CHECK FOR INCREASING ORDER BOOK

140 IF(OCF(L,N) .LT. OCFNOM * R(L,J)) GO TO 150
IF(FLAG.EQ.0) GO TO 60

CHECK FOR 1ST QUARTER UNDERLOAD

150 IF(N.GT.2) GO TO 155
IF(PA(L,N) .LE. COL(L,N) + IFIX(ORF(L,N) * (13 - OCFMIN)/13.0)) GO TO 155
FES(L,N) = FES(L,NN) * COL(L,N) + IFIX(ORF(L,N) * (13 - OCFMIN)/13.0)
GO TO 120

CHECK FOR MINIMUM ORDER BOOK

155 IF(OCF(L,N) .GT. OCFMIN * ORF(L,N) / 13.0) GO TO 160
FES(L,N) = FES(L,NN) * OCFMIN * ORF(L,N) / 13.0 - OCF(L,N)
PA(L,N) = OFP(L,N) - FES(L,N) + FES(L,NN)
OCF(L,N) = OCF(L,NN) + ORF(L,N) * PA(L,N)

160 RR(L,N) = R(L,J)

162 IF(N, GT. 3) GO TO 165
CUMDIF = CUMDIF + ONP(L,N) - ONP(L,N)

165 IF(N, LT. 7) GO TO 30
200 CONTINUE
RETURN
170 RETURN
END
The subroutine QPLANPRINT serves two purposes:

a) to portray the new quarterly production plans resulting from the subroutine QPLAN;

b) to present the information in a readily readable form, which includes a calculation of the actual order load in weeks. This data is not required as part of the model logic, but completes the plan presentation format as used within the Company and permits diagnostic analyses.

In addition to the presentation of the quarterly based file contents, QPLANPRINT also identifies the parameters applicable to the quarterly plan calculation.
SUBROUTINE QPLANPRINT(ORF, COL, FES, WIP, IONP, R, AFSMAX, OCFMIN, OCFNOM, 10NP, PA, OCF, OPF, RR, AF, IFILE, IFILE)

REAL AF FILE(25, 3)
INTEGER ORF(10, 7), COL(10, 7), FES(10, 7), WIP(10, 7), IONP(10, 7), R(10, 10)
AFSMAX(10), OCFMIN, OCFNOM, ONP(10, 7), PA(10, 10), OCF(10, 7), OPF(10, 7)
2 RR(10, 11), PASUM, OCFWKS(7), IFILE(25, 23)

DO 100 K = 1, 10
IF (IFILE(K, 11), NE: 1) GO TO 100
WRITE (2, 30) AF FILE(K, 1)

30 FORMAT(1H1, ////6X 'QUARTERLY PLAN FOR PRODUCT TYPE', 2X, 1A8 //</>)

CALCULATE WEEKS WORTH OF ORDERS CARRIED FORWARD

COL(K, 2) = COL(K, 2) + COL(K, 1)
DO 60 N = 1, 7
I = N
PASUM = 0
40 I = I + 1
IF (I, LE: 7) GO TO 50
PA(K, I) = PA(K, I - 1)
50 PASUM = PASUM + PA(K, I)
IF (OCF(K, N), LE: PASUM) GO TO 60
IF (I, LT: 10) GO TO 40
PASUM = OCF(K, N)
GO TO 70
60 PASUM = PASUM - PA(K, I)
70 IF (PA(K, I), GT: 0) GO TO 75
N = 1
75 M = PA(K, I)
80 OCFWKS(N) = (I - N - 1) * 13 + (OCF(K, N) - PASUM) * 13/M

WRITE REPORT

WRITE (2, 310) (OCF(K, I), I = 1, 7), (OCFWKS(I), I = 1, 7), (COL(K, I), I = 1, 7), (0
1RR(K, I), I = 2, 7), (PA(K, I), I = 2, 7), (FES(K, I), I = 1, 7), (OPF(K, I), I = 2, 7), (2
WIP(K, I), I = 1, I = 177), (RC(K, I), I = 2, 7), (ONP(K, I), I = 2, 7)

310 FORMAT(1H0, 6X, 'QUARTER', 21X 'I', 5X 'F', 5X 'Q1', 5X 'Q2', 5X 'Q3', 5X 'Q4', 15X 'Q5', 5X 'Q6', 5X 'Q7', 6X, 'ORDERS CARRIED FORWARD', 4X, 717//6X, 'ORDER BOOK
2 (WEEKS)', 8X, 717//6X, 'CURRENT ORDER LOAD', 8X, 717//6X, 'ORDERS RECEIVED
67//)
WRITE (2, 320) (IONP(K, J), J = 2, 4), (AFSMAX(K), OCFMIN, OCFNOM, (R(K, J), J = 1, 110)
317//36X, 517)

100 CONTINUE
RETURN.
END
QUARTERLY PLAN FOR PRODUCT TYPE MF6AM 02

<table>
<thead>
<tr>
<th>QUARTER</th>
<th>B/F</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORDERS CARRIED FORWARD</td>
<td>310</td>
<td>335</td>
<td>372</td>
<td>347</td>
<td>310</td>
<td>262</td>
<td>212</td>
</tr>
<tr>
<td>ORDER BOOK (WEEKS)</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>CURRENT ORDER LOAD</td>
<td>25</td>
<td>285</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ORDERS RECEIVED FORECAST</td>
<td>518</td>
<td>507</td>
<td>495</td>
<td>483</td>
<td>472</td>
<td>461</td>
<td></td>
</tr>
<tr>
<td>PRODUCTION TO BE ALLOCATED</td>
<td>463</td>
<td>520</td>
<td>520</td>
<td>520</td>
<td>520</td>
<td>511</td>
<td></td>
</tr>
<tr>
<td>STOCK</td>
<td>0</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>86</td>
</tr>
<tr>
<td>OFF-LINE PLAN</td>
<td>520</td>
<td>520</td>
<td>520</td>
<td>520</td>
<td>520</td>
<td>520</td>
<td></td>
</tr>
<tr>
<td>WORK IN PROGRESS</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>LINE RUNNING RATE</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>MATERIAL PLAN</td>
<td>520</td>
<td>520</td>
<td>520</td>
<td>520</td>
<td>520</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTES:

PREVIOUS MATERIAL PLAN: 520 520 520
MAXIMUM AUTHORISED STOCK: 154
MINIMUM ORDER BOOK: 6
NOMINAL ORDER BOOK: 10
RECOMMENDED RUNNING RATES: 0 2 5 12 18

Sample Product Quarterly Plan
Subroutine QTREVENTS controls the sequence of the quarterly activities, the majority of which are dependent modules.

a) Dependent modules

- **QTRRESET** - resets the planning and execution files in preparation for the next quarterly cycle.
- **FORECAST** - calculates the quarterly orders received forecast from the order received history.
- **QPLAN** - establishes a new quarterly plan in response to the new orders received forecast.
- **QPLANPRINT** - optional module permitting the result of the planning process to be examined.
- **WEEKLYPLAN** - converts the time base from quarterly to weekly planning data.
- **EXPLODE** - generates nett requirements for lower level parts by using the product structure relationships.
- **SCHED** - establishes the supplier schedule for component items.
- **RECEIVE** - determines the actual input pattern of material against the planned supplier schedule.

b) Independent modules

- **TRENDI** - modifies the product mix data when orders received trends are introduced into the model.
- **EBQ** - calculates the "economic batch quantity" to be used in the sub-assembly programme determination.
CALL QTREVENTS
RESET SYSTEM FILES
RECALCULATE PRODUCT MIX TREND
ESTABLISH NEW FORECAST
ESTABLISH PRODUCTION PLAN
PRINT PLAN?
Y
PRINT PLAN
ESTABLISH WEEKLY PLAN
CALCULATE MATERIAL REQUIREMENTS
ESTABLISH BATCH SIZES
ESTABLISH SUPPLIER SCHEDULES
DETERMINE MATERIAL INPUT PATTERN
RETURN

SEGMENT NAME
QTRESET
TREND I.
FORECAST
QPLAN
QPLANPRINT
WEEKLYPLAN
EXPLODE
EBQ
SCHED
RECEIVE

Summarised Flow Chart - Subroutine Qtrevents
QTREVENTS

SUBROUTINE QTREVENTS(ORDHIST, AFILE, IFILE, IPRINT, COL, ONP, R, RESPONSE
1; ISEED1; TABLE1; ISEED2; TABLE2; OCFMIN, OCFNOM, SCHEDULE, INPUT, OPFILE, O
2; DBK, TOTAL, WEEK, REFWEEK, TQTR, EXFILE, REQFILE, DELPLAN, NETPLAN, ICOUNT
3; ISHORT; CAPACITY, UTIL, LEADTM, TREND, TABLE6, PRODMIX, ALLQ, MODE, EQUPERF
4; IFITEMPERF, RQPERF, EQUBUFR)

REAL AFILE(25, 3), TRENP(7), PRODMIX(7)
INTEGER IFILE(25, 23), ORDBK(5, 12), ORF(10, 7), COL(10, 7); FES(10, 7)
1WIP(10, 7), IONP(10, 7); R(10, 10), AFMAX(10), OCFMIN, OCFNOM, IONP(10, 7), P
2A(10, 10); OCF(10, 7), OFP(10, 7); RR(10, 11), REQFILE(25, 52), DELPLAN(5, 5
32), CAPACITY, RESPONSE, TABLE1(7, 2), TABLE2(7, 2); ISEED1, ISEED2, SCHED
4ULE(25, 13), INPUT(25, 52), OCF(10, 10), OCF(10, 10), OFP(10, 10), RR(10, 11), REQFILE(25, 52), NETPLAN
5(5, 52), WEEK, REFWEEK, EXFILE(25, 52), LEADTM(5, 52), TABLE
66(7, 2), ALLQ(12, 50, 6), EQUPERF(10, 30), IFITEMPERF(10, 30), RQPERF(2, 30)
7; ISHORT(25), ISHORT(25), EQUBUFR

IF(IPRINT.GT.1) GO TO 5
WRITE(2, 3)
3 FORMAT(1HO, 6X, * * * * QTREVENTS * * * * *1//)
5 CALL QTRRESET(OPFILE, ORDBK, TOTAL, WEEK, REFWEEK, TQTR, EXFILE, IFILE, AFILE, COL, SCHEDULE, INPUT, NETPLAN, IONP, OFF, IPRINT, ICOUNT, ISHORT, LEADTM, OCF, IFITEMPERF, RQPERF)
2 CALL TREND1(TABLE6, PRODMIX, TREND, IPRINT)
2 CALL FORECAST(ORDHIST, ORF, IPRINT, AFILE, IFILE)
2 CALL QPLANCORF; COL, FES, WIP, IONP, R, AFMAX, OCFMIN, OCFNOM, IONP, PA, OCF, IPRINT, RR, AFILE, REQFILE, IPRINT, MODE, EQUBUFR)
10 IF(IPRINT.GT.5) DO 10
CALL QPLANPRINT(CORF, COL, FES, WIP, IONP, R, AFMAX, OCFMIN, OCFNOM, IONP, PA, OCF, OFP, RR, AFILE, IFILE)
10 CALL WEEKLYPLAW(OFP, REQFILE, IFILE, PA, DELPLAN, IPRINT, AFILE)
2 CALL EXPLODE(AFILE, IFILE, REQFILE, IPRINT, ALLQ)
2 CALL EBQ(REQFILE, IFILE, IPRINT, AFILE, CAPACITY, UTIL)
2 CALL SCHED(REQFILE, SCHEDULE, RESPONSE, IFILE, AFILE, IPRINT)
2 CALL RECEIVE(SCHEDULE, INPUT, ISEED1, TABLE1, ISEED2, TABLE2, IFILE, IPRINT)
100 RETURN
END
The prime planning files are reset each quarter by subroutine QTRRESET, thus limiting the amount of redundant file space in the model. QTRRESET also initialises a number of files and parameters used for performance measurement and diagnostic analysis.

The quarter counter is reset to 13.0 weeks and the component service level status variables are initialised in preparation for the next quarterly cycle. For the first simulation cycle only, the off-line plan, OFP, is loaded with the material on-line plan, ONP. The parameter REFWEEK is incremented by 13 weeks, indicating the last week number of the previous quarterly period.

The following files are then reset as indicated.

a) ORDBK

The equipment order book cannot be simply advanced by thirteen weeks, since the overdue orders are still required to be serviced. In the interests of file economy, these records are transferred to the "tail end" of the file and identified with an "overdue" flag as shown.

[Diagram showing resetting of ORDBK]
b) ONP

The material plan, which is carried forward to QPLAN as a plan constraint, is reset by moving the data forward by one period.

c) TOTAL, NETPLAN, INPUT, EXFILE

The total load, loading plan, material input and expedite files are each organised into 52 weekly periods. In each case the overdue periods are discarded and the file contents advanced by 13 periods.

d) SCHEDULE

The schedule file contains twelve monthly periods plus a gross overdue, or arrears, value. The residue of the current quarter is added to the arrears total and the file advanced in three periods increments.

e) The current order load is derived by searching ORDBK for undelivered quantities and loading these into the file COL for subsequent use by subroutine QPLAN.

f) The files containing order performance statistics are cleared down in preparation for the next quarterly cycle.
Summarised Flow Chart - Sub-routine Qtrreset
SUBROUTINE QTRRESET(OPFILE, ORDBK, TOTAL, WEEK, REFWEEK, TQTR, EXFILE, IFILE, AFIL, COL, SCHEDULE, INPUT, NETPLAN, ONP, OFP, IPRINT, ICOUNT, ISHORT, ELSEADTH, EQUPERF, ITEMPERF, ORDPERF)

REAL AFIL(25,3)
INTEGER IFILE(25,23), OPFILE(125,5,10), ORDBK(5,52,20,5), TOTAL(5,152), WEEK, EXFILE(25,52), COL(10,7), REFWEEK, SCHEDULE(25,13), INPUT(225,52), NETPLAN(5,52), OFP(10,7), ONP(10,7), IHOLD(5), LEADTH(5,52), EQP(10,30), ITEMPERF(10,30), ORDPERF(2,30), ICOUNT(25), ISHORT(25)

IF (IPRINT.GT.1) GO TO 3
WRITE(2,1)
1 FORMAT(1HO,6X,1* * * * QTRRESET * * * * *'/))

TQTR=13.0
WEEK=1
ICOUNT(1)=1
ISHORT(1)=0
DO 4 I=1,5
DO 4 J=1,52
4 LEADTH(I,J)=0
IF(REFWEEK.GT.0) GO TO 5
DO 243 N=1,5
DO 243 J=1,7
5 REFWEEK=REFWEEK+13

RESET ORDER BOOK

DO 80 I=1,5
IF(IFILE(I,11).GT.1) GO TO 80
DO 80 MA=1,13
MF=MA+39
MG=MA+26
DO 80 MD=1,20
DO 80 ME=1,5
50 IHOLD(ME)=ORDBK(I,MA,MD,ME)
DO 60 MB=MA,MG,13
MC=MB+13
DO 60 ME=1,5
ORDBK(I,MB,MD,ME)=ORDBK(I,MC,MD,ME)
60 CONTINUE
DO 70 ME=1,5
70 ORDBK(I,MF,MD,ME)=IHOLD(ME)
IF(ORDBK(I,MF,MD,3).EQ.0) GO TO 80
ORDBK(I,MF,MD,4)=1
80 CONTINUE

RESET ON LINE PLAN

DO 120 I=1,10
DO 120 J=1,6
K=J+1
120 ONP(I,J)=ONP(I,K)

RESET TOTAL LOAD, EXPEDITE FILE AND INPUT FILE

DO 174 I=1,5
DO 170 J=1,39

- 424 -
K=J+13
NETPLAN(I,J)=NETPLAN(I,K)
170 TOTAL(I,J)=TOTAL(I,K)
DO 172 J=40,52,
NETPLAN(I,J)=0
172 TOTAL(I,J)=0
174 CONTINUE
IF(IPRINT.GT.3)GO TO 175
WRITE(2,110)REFWEEK
110 FORMAT(91,6X,IFILES RESET AT END OF QUARTER'/6X,'REFWEEK'=',2I4)
CALL ORDBKPRT(ORDBK,AFILE,IFILE,TOTAL,REFWEEK)
175 IF(IPRINT.GT.4)GO TO 179
WRITE(2,176)
176 FORMAT(90,6X,EXEDITING FREQUENCY'/6X,'I=1,25
IF(IFILE(I,11).NE.4)GO TO 178
WRITE(2,177)AFILE(171),INPUT(I,J),J=1,20),EXFILE(I,J),J=1,20)
177 FORMAT(94,6X,AB,20,NODES
178 CONTINUE
179 DO 190 J=1,25
IF(IFILE(I,11).NE.4)GO TO 190
DO 180 J=1,39,
K=J+13
INPUT(I,J)=INPUT(I,K)
180 EXFILE(I,J)=EXFILE(I,K)
DO 183 J=40,52,
INPUT(I,J)=0
EXFILE(I,J)=0
183 CONTINUE
C C
RESET SCHEDULE FILE
DO 185 J=2,4
185 SCHEDULE(I,1)=SCHEDULE(I,1)*SCHEDULE(I,J)
DO 186 L=2,10,
K=L+3
186 SCHEDULE(I,K)=SCHEDULE(I,1)
IF(IPRINT.GT.3)GO TO 190
WRITE(2,187)AFILE(171),INPUT(I,J),J=1,20),EXFILE(I,J),J=1,20)
187 FORMAT(96,6X,PRODUCT.../bin/8X,'PRODUCT ',8X,2015/14X,2015)
190 CONTINUE
C C
CALCULATE CURRENT ORDER LOAD
200 IF(IPRINT.GT.3)GO TO 205
WRITE(2,203)(J,J=1,6)
203 FORMAT(98,6X,'CURRENT ORDER LOAD',2X,'0/D',3X,6(3X,'QF11))
205 DO 250 I=1,5
IF(IFILE(I,11).NE.1)GO TO 250
COL(I,1)=O
DO 210 L=1,52
DO 210 H=1,20
IF(ORDBK(I,L,H,M,4).EQ.0)GO TO 210
COL(I,1)=COL(I,H)*ORDBK(I,L,H,3)
210 CONTINUE
DO 230 J=2,5
COL(I,J)=0
K=(J-1)*13
N=K-12
DO 230 L=N,K  
DO 230 M=1,20  
IF(ORDBK(I,L,M,4),EQ,1) GO TO 230  
COL(I,J)=COL(I,J)*ORDBK(I,L,M,3)  
230 CONTINUE  
IF(IPRINT,GT,3) GO TO 250  
WRITE(2,220)FILE(I,J),(COL(I,J),J=1,7)  
220 FORMAT(1HO,10X,A8,6X,15,3X,6I5)  
250 CONTINUE  
CONTINUE  
RETURN  
END  

RESTT DELIVERY PERFORMANCE STATISTICS  

DO 260 I=1,10  
DO 260 J=1,30  
EQUIPERF(I,J)=0  
ITEMPERF(I,J)=0  
260 CONTINUE  
DO 270 I=1,2  
DO 270 J=1,30  
ORDPERF(I,J)=0  
270 CONTINUE  
RETURN  
END
The generation of pseudo-random number streams is required to support the various sampling tables and distributions accessed by the model.

Two factors were considered when selecting a suitable pseudo-random number generator design:

- the generator should be capable of accepting an external seed, which could be returned for future use. This facilitates the identification of a unique stream of seeds with a particular sampling facility.

- the generator should be capable of accepting negative seeds if antithetic sequences are required.

The design chosen is a multiplicative congruential method as described by Naylor et al (1966). The model is designed to run on an ICL1904S computer, which is a 24 bit word machine.

The number of bits available for number definition is 23.

The modulus, \( m = 2^b \), where \( b \) is the number of binary digits (bits) in a word. This \( m = 2^b = 2^{24} \).

The value of "\( a \)" which is relatively prime to "\( m \)" is given by the congruence relation

\[ a \equiv \pm 3 \pmod{8} \]

which may be expressed as

\[ a = 8t \pm 3 \]

where "\( t \)" is any positive integer. "\( t \)" should be chosen to give "\( a \)" close to \( 2^{b/2} \), or 2048 (for a 24 bit machine). Hence a value of \( 2048 - 3 = 2045 \) was selected.
A starting value $n_0$ (the seed) is selected, which should be relatively prime to $2^b$, a condition which is satisfied by any odd number.

The product $a_n$ will consist of $2b$ bits, the lower-order "$b" bits representing $n_1$. This value is returned as a seed for subsequent use. The subroutine returns $r_1$, which is a uniformly distributed variate defined on the unit interval, where $r_1 = n_1/2^b = n_1/2^{23}$

The subroutine will produce a series of random variates $r_1 \ldots r_n$ which will repeat after $2^{b-2} = 2097152$ numbers.

The subroutine depends upon the standard FORTRAN compiler logic, where an integer multiplication instruction automatically discards the high order $b$ bits. The compiler "XFAT" will not normally allow this condition (which results in the overflow register being set), thus an error trap has been introduced to suppress an error exit on overflow condition. The error condition is re-established on leaving the sub-routine.

The subroutine is based on an IBM library program modified to comply with the above parameter definitions. A sequence to accommodate negative seeds has been included to provide the facility of antithetic streams.

The output of the pseudo-random generator has not been tested by any of the standard statistical methods. Sequences of numbers have been visually checked as a uniform distribution and after application in a random normal deviate generator. Both instances provide acceptable results in the context of the model. If a rigorous analysis of results is required, including meaningful tests of significance, appropriate statistical tests must be applied.
RANDOM

SUBROUTINE RANDOM(E,J)
EXTERNAL OVER
CALL FTRAP(OVER)
IF(J).EQ.60,40,10
10 J=2045*J
 IF(J).GT.40,30
20 J=J+8388607+1
30 E=J/8388608.0
 CALL FRESH
 RETURN
40 WRITE(2,50)
50 FORMAT(1HO,6X,'ZERO RANDOM STREAM')
 STOP
60 J=2045*J
 IF(J).GT.40,70
70 J=-8388607-1
80 E=1.0+J/8388608.0
 CALL FRESH
 RETURN
END

ERROR TRAP

ERROR TRAP
SUBROUTINE OVER(I)
 IF(I,EQ.50)GO TO 100
 WRITE(2,10)I
10 FORMAT(1HO,6X,'EXECUTION ERROR -',I3)
 PAUSE EE
100 RETURN
END
The schedules of planned deliveries from the supplier are established in subroutine SCHED and maintained in monthly buckets in file SCHEDULE. The actual deliveries will tend to conform to a pattern around the planned receipt date. Subroutine RECEIVE establishes the deviation from the planned receipt date and creates a new file, INPUT, which contains the weekly programme of actual material receipts.

The schedule response time, which is the period during which no changes may be made to the supplier schedule, is also applied to the input file. This ensures that the two files, SCHEDULE and INPUT are consistent. The response time is stored in months for use in SCHED and is converted to an equivalent weekly value for use in RECEIVE.

The schedule of supplier deliveries is maintained in SCHEDULE as a series of monthly quantities due in the last week of the month. Subroutine RECEIVE determines, for each scheduled receipt, the actual delivery week. Two sampling tables are used to determine the spread around the scheduled date; the first is applicable to normal deliveries beyond the schedule response time, and the second is applicable to "arrears" to schedule.

The file INPUT is re-created each quarter following establishment of the file SCHEDULE in subroutine SCHED. All data in file INPUT is cleared down beyond the schedule response time and the data within the response period is carried forward. No further changes to the data currently within this period are permissible.

File SCHEDULE is interrogated for finite scheduled receipts. The actual receipt date of the batch is determined by sampling from TABLE 1, which is a frequency histogram defining the deviation from the planned date and is of the form shown in (a). The relative frequency for each interval is shown in (b).
Thus, the actual delivery date is equally distributed around the planned date with a maximum deviation of 3 weeks. The scheduled batch is assumed to be delivered in one shipment, thus the scheduled quantity may be loaded into the appropriate week in file INPUT.

Within the schedule response time, the gross value of schedule receipts is compared to the actual input batches in file INPUT for the same period to arrive at a net value. This net value is treated as a schedule arrear. Arrears inputs augment the normal receipt batches, the receipt dates being derived by sampling from TABLE 2. TABLE 2 is a cumulative frequency histogram defining the actual delivery week as shown below.
File INPUT is, therefore, created from three elements:

a) Previous INPUT data, within the response period;

b) Arrears to schedule, being the difference between the schedule batches and the actual input within the response period;

c) Normal schedule receipts beyond the response period.
SUBROUTINE RECEIVE(SCHEDULE, INPUT, ISEED1, TABLE1, ISEED2, TABLE2, IFILE, IPRINT, RESPONSE, AFILE)

FIRST BUCKET OF SCHEDULE IS ARREARS
EIGHT WEEKS OF INPUT IS FIXED

REAL AFILE(25, 3)
INTEGER INPUT(25, 52), SCHEDULE(25, 13), TABLE1(7, 2), TABLE2(7, 2), IFILE(25, 23), RESPONSE

IF(IPRINT.GT.1) GO TO 400
WRITE(2, 420) 420 FORMAT(1HO, 6X, '******** RECEIVED ********')
400 M = RESPONSE*4
IF(IPRINT.GT.3) GO TO 1
WRITE(2, 200)
200 FORMAT(1H0//6X, 'MATERIAL INPUT')
1 DO 70 N = 1, 25
   IF(IIFILE(N, 11) .NE. 4) GO TO 70
   DO 5 J = N+1, 52
   5 INPUT(N, J) = 0
   NETVAL = 0
   DO 10 J = 1, RESPONSE+1
      NETVAL = NETVAL + SCHEDULE(N, J)
   10 J = N+1
   15 NETVAL = NETVAL - INPUT(N, J)
   IF(IPRINT.GT.1) GO TO 15
   WRITE(2, 100) NETVAL
   100 FORMAT(1HO, 6X, 'NETVAL = ', I6)
   16 DO 30 J = RESPONSE+2, 13
      K = 4*(J-1)-3
      IF(SCHEDULE(N, J) .EQ. 0) GO TO 30
      NUMBER = 0
      IF(ISEED1 .EQ. 0) GO TO 20
      CALL HSAMPLE(ISEED1, NUMBER, TABLE1)
      IF((K+NUMBER) .LE. 52) GO TO 20
      NUMBER = 52 - K
   20 INPUT(N, K+NUMBER) = INPUT(N, K+NUMBER) + SCHEDULE(N, J)
   30 CONTINUE
   IF(NETVAL) 40, 65, 35
   35 NUMBER = 2
   IF(ISEED2 .EQ. 0) GO TO 37
   CALL HSAMPLE(ISEED2, NUMBER, TABLE2)
   37 I = NUMBER + M
   GO TO 60
50 INPUT(N, I) = 0
55 IF(I, GT. 52) GO TO 65
60 INPUT(N, I) = INPUT(N, I) + NETVAL
   65 IF(IPRINT.GT.3) GO TO 70
   WRITE(2, 300) AFILE(N, I), (INPUT(N, J), J = 1, 40)
300 FORMAT(1HO, 6X, A8, 2X, '2015////16X, 2015')
70 CONTINUE
RETURN
END
The subroutine SCHED converts the requirements plan for a purchased part into a monthly "call-off" schedule for the supplier.

The prime input to the subroutine is the net requirement for the part, which is presented to the program in weekly detail. The previous call-off schedule is used as a reference, with any backlog appearing as a single "arrears" value. Early deliveries will appear as a negative arrear to schedule.

The existing scheduling system logic employs an eight week "frozen period", or schedule response time, during which the previous schedule is maintained. Any prospective change to the schedule thus applies immediately beyond the fixed period.

In practice, any changes required during the fixed period and any significant changes beyond the fixed period, would be discussed between the Purchasing Department and the supplier and a schedule revision is agreed. The final schedule is input to the system and is used to monitor supplier performance. No formal procedure is present to feed back the revised information and evaluate the implications on the plan.

The simulation model uses an input parameter to determine the response time before schedule changes are effected and assumes that:

a) the "frozen" period is inviolable;

b) no manual intervention may take place;

c) purchase order cover is available to support the required call-off schedule.
The final schedule returned to the model will thus reflect:

(i) the previous schedule arrears;

(ii) the first "x" call-offs of the previous schedule, where x is the response time in months;

(iii) the following new requirement in monthly periods based on the net requirements.
SUBROUTINE SCHED(REQFILE,SCHEDULE,RESPONSE,IFILE,AFILE,IPRINT)
REAL AFILE(25,3)
INTEGER NETPLAN(12),OLDSCHED(13),SCHEDULE(25,13),RESPONSE,REQFILE(125,52),IFILE(25,23)

NOTE THAT FIRST PERIOD OF SCHEDULE IS ARREARS

IF(IPRINT.GT.1)GO TO 400
WRITE(2,420)

420 FORMAT(1HO,6X,'* * * SCHED * * * */)  
400 IF(IPRINT.GT.3)GO TO 1
WRITE(2,150)(J,J=1,12)

150 FORMAT(1HO//6X,'COMPONENT SCHEDULES'//16X,'ARREARS',121B)
  DO 70 N=1,25
    IF(IFILE(N,11).NE;4)GO TO 70
    WRITE(2,250)AFILE(N,1),SCHEDULE(N,J),J=1,13

250 FORMAT(1HO/6X,'PART NUMBER',12X,'OLD SCHED',18,2X,12IB)
  DO 3  J=1,13
    OLDSCHED(J)=SCHEDULE(N,J)

  DO 5 J=1,12
    NETPLAN(J)=0
    M=4*J-3
    DO 5 K=M,M+3

5     NETPLAN(J)=NETPLAN(J)+REQFILE(N,K)
      SCHEDULE(N,1)=OLDSCHED(1)
      NETVAL=OLDSCHED(1)
      IF(RESPONSE.EQ.0)GO TO 20
      DO 10 J=2,RESPONSE+1
        IF(J.GT.12)GO TO 70
        NETVAL=NETVAL+OLDSCHED(J)-NETPLAN(J-1)

10     SCHEDULE(N,J)=OLDSCHED(J)
     20 K=RESPONSE
     30 K=K+1
     IF(NETVAL.GT.NEJPAN(K))GO TO 40
      SCHEDULE(N,K+1)=NETPLAN(K)-NETVAL
      GO TO 50
     40 NETVAL=NETVAL-NETPLAN(K)
      SCHEDULE(N,K+1)=0
      IF(K.LT.12)GO TO 30
     50 IF(K.GE.12)GO TO 70
      DO 60 L=K+1,11

60     SCHEDULE(N,L+1)=NETPLAN(L)
      IF(IPRINT.GT.3)GO TO 70
      WRITE(2,350)(SCHEDULE(N,J),J=1,13)
     350 FORMAT(1HO/6X,'NEW SCHED',18,2X,12IB)
     70 CONTINUE
     80 RETURN
END
Orders which have been selected in subroutine ALLOCATE are moved from "open" to "allocation" status in file OPFILE and have their component requirements allocated against stock. This is performed by subroutine STKALL.

Entry into the subroutine is followed by a check that a free record is available in the allocate queue, ALLQ, for the required product. If the file is full, control is returned to the calling programme.

The allocation queue record is created by writing into the next available space the order number, item number, due date and quantity. The minimum lead time that the order must remain in ALLQ for material preparation is determined by a random normal deviate derived from the calling program. The shop order number, which links quantities of the same product, is also derived from the calling programme.

For each product requirement entering the subroutine, the components are allocated by extending the quantity required by the component relationships contained within IFILE. The allocation procedure consists of incrementing the "allocated stock" field in IFILE for each component by the gross requirement. Each requirement will increment a "required" counter, ICOUNT, and each shortage encountered (where allocated stock exceeds physical stocks) will increment the "shortage" counter, ISHORT. These status variables are subsequently used for service level reporting.

Where a non-stocked item (identified as "not to be issued" in IFILE) is encountered, a further level of allocation is performed on the components of the non-stocked item.
Summarised Flow Chart - Subroutine Stkall
SUBROUTINE STKALL(IFILE, QUANTITY, N, ALLQ, ORDNO, ITEMNO, DDATE, OUTPUT
/ICOUNT,ISHORT,IPRINT,IMAX,AFILE,*)
REAL AFILE(25,3)
INTEGER QUANTITY, IFILE(25,23), REQD, HOLD(7,2), ALLQ(12,5076),
1 DDATE, ORDNO, ICOUNT(25), ISHORT(25)
IF(IPRINT.GT.1)GO TO 3
WRITE(2,1)
1 FORMAT(1HO,6X,** ** ** STKALL ** ** **://)
3 K=0
IF(IPRINT.GT.1)GO TO 5
WRITE(2,20)ORDNO,ITEMNO,DDATE,QUANTITY
20 FORMAT(1HO,6X,ORDNO='1',16,2X,ITEMNO='1',16,2X,DDDATE='1',16,2X,QU
1 ANITY='1',16)
C
FIND NEXT SPACE IN QUEUE
5 DO 10 J=1,50
IF(ALLQ(N,J,4).EQ.'0')GO TO 30
10 CONTINUE
IF(IPRINT.GT.5)GO TO 140
WRITE(2,38)AFILE(N,1)
38 FORMAT(1HO,6X,'EXCEEDED ALLOCATION QUEUE SIZE FOR PRODUCT No.1,2X;
1 A8)
GO TO 140
C
MOVE ITEM INTO ALLOCATION QUEUE
30 ALLQ(N,J,1)=ORDNO
ALLQ(N,J,2)=ITEMNO
ALLQ(N,J,3)=DDATE
ALLQ(N,J,4)=QUANTITY
35 ALLQ(N,J,5)=NINT(OUTPUT)
ALLQ(N,J,6)=IMAX
IF(IPRINT.GT.2)GO TO 40
WRITE(2,38)AFILE(N,1),(ALLQ(N,J,M),M=1,6)
38 FORMAT(1HO,6X,'ITEM ADDED TO ALLQ FOR PRODUCT',2X,A8,6X,'ORDNO=',
1 16,2X,'ITEMNO=',16,2X,'DDDATE=',16,2X,'QUANTITY=',16,2X,'TIME=',16,2X,'IS
2 HOP ORD=',16)
C
ALLOCATE COMPONENTS
40 DO 110 J=1,972
IF(IFILE(N,J).EQ.0)GO TO 110
L=IFILE(N,J)
ICOUNT(L)=ICOUNT(L)+1
IP=J+1
REQD=QUANTITY*IFILE(N,IP)
IF(IFILE(L,15).EQ.1)GO TO 50
IFILE(L,17)=IFILE(L,17)+REQD
IF(IFILE(L,16).EQ.IFILE(L,17))GO TO 110
ISHORT(L)=ISHORT(L)+1
GO TO 110
50 K=K+1
HOLD(K,1)=L
HOLD(K,2)=REQD
110 CONTINUE
IF(K.EQ.0)GO TO 120
N=HOLD(K,1)
QUANTITY=HOLD(K,2)
K=K-1
GO TO 40
120 RETURN
140 RETURN 1
END
One of the key performance measures required of the model is to monitor the investment in inventory. The evaluation of inventory across the various production stages is executed within the subroutine STOCKVAL.

Each product standard cost comprises four elements; direct material, material overheads, direct labour and labour overheads. The overhead rates selected are fairly representative of the composition of a typical product. Material overheads are applied (or recovered) at the point of issue to work in progress at a rate of 20% of the standard cost of material. Labour overheads are applied at the point of transfer to stock (or finished goods warehouse) at a rate of 250% of direct labour.

Values held in IFILE are prime costs only, nett of overhead elements. At each level of assembly, the accumulated value of prime costs, including the labour content for completion of the assembly as a stockable item, is held. If "M" is the accumulated value of direct material and "L" is the corresponding value of direct labour, the total standard cost, "S", for the assembly is

\[ S = 1.2M + 3.5L \quad (1) \]

Figures A/B below illustrate the assumptions used to arrive at a reasonable approximation of the added value as the production process advances.
The formulae used to value each category of inventory are derived from the above profiles.

**COMPONENTS**

The value of stock is taken as the standard cost of material less any material overhead contribution. The total stock value is the product of the standard cost and the physical stock for each component item.

**SUB-ASSEMBLIES**

Sub-assemblies held in stock are assumed to be fully completed including assembly and test labour. Thus, the standard value of each item is given by the relationship in (1) above.

As sub-assemblies progress through work in progress, labour is assumed to be applied to the material linearly. The standard value is therefore given by the relationship

\[ S = 1.2M + 1.75L \] (2)
EQUIPMENT

Equipment is valued at four levels. Stock which is ready for despatch (commercial stock) is valued at the full standard cost as given in (1) above.

Equipment in final test is assumed to have accumulated 95% of the total labour effort, giving a standard value of

\[ S = 1.2M + 3.3L \]  

Equipment held in finished product stock is normally nett of final test labour, and has been assumed to have accumulated 85% of the labour value. Thus, finished equipment stock is valued as

\[ S = 1.2M + 3.0L \]  

Equipment in the product assembly stage will have accumulated the labour value of the comprising sub-assemblies plus a proportion of the final assembly labour. This is assumed to be 70% of the full labour content, giving a standard value of

\[ S = 1.2M + 2.5L \]  

All values except commercial stock are derived from the item master record IFILE. Commercial stock is determined by scanning the product record in the orders placed file OPFILE and summing the items in "despatch" status.

The sum of each of the above stock categories indicates the total inventory investment for the period.
Summarised Flow Chart - Sub-routine Stockval
SUBROUTINE STOCKVAL(IFILE, MONTH, OPFILE)

INTEGER COMPVAL, WIPVAL, SUBVAL, FESVAL, IFILE(25, 23), TOTVAL, CSVAL, OPFILE(125, 5, 5, 10), TESTWIP

CSVAL = 0
COMPVAL = 0
WIPVAL = 0
SUBVAL = 0
FESVAL = 0
TESTWIP = 0
DO 100 I = 1, 25
IF = IFILE(I, 11)
GO TO (10, 20, 30); IF
10 FESVAL = FESVAL + IFILE(I, 16) * (IFILE(I, 13) * 1.2 + IFILE(I, 14) * 3.0)
WIPVAL = WIPVAL + IFILE(I, 18) * (IFILE(I, 13) * 1.2 + IFILE(I, 14) * 2.5)
TESTWIP = TESTWIP + IFILE(I, 19) * (IFILE(I, 13) * 1.2 + IFILE(I, 14) * 3.3)
GO TO 100
20 SUBVAL = SUBVAL + IFILE(I, 16) * (IFILE(I, 13) * 1.2 + IFILE(I, 14) * 3.5)
WIPVAL = WIPVAL + IFILE(I, 18) * (IFILE(I, 13) * 1.2 + IFILE(I, 14) * 2.5)
GO TO 100
30 COMPVAL = COMPVAL + IFILE(I, 16) * IFILE(I, 13)
100 CONTINUE
DO 150 I = 1, 125
DO 150 J = 1, 5
DO 150 K = 1, 5
IF (OPFILE(I, J, K, 9).EQ.0) GO TO 150
L = OPFILE(I, J, K, 1)
CSVAL = CSVAL + OPFILE(I, J, K, 9) * (IFILE(L, 13) * 1.2 + IFILE(L, 14) * 3.5)
150 CONTINUE

TOTVAL = COMPVAL + SUBVAL + WIPVAL + FESVAL + CSVAL + TESTWIP
WRITE (2, 200) MONTH, COMPVAL, SUBVAL, WIPVAL, FESVAL, TESTWIP, CSVAL, TOTVAL
RETURN
END
The requirement for sub-assemblies is derived each quarter from the requirements calculation contained in subroutine EXPLODE. Since the recalculation is executed infrequently, some mechanism is required to review and modify the sub-assembly schedule according to the actual demand pattern and stock profile. This re-balancing is performed in subroutine SUBPROG.

The role of SUBPROG is twofold; to convert the requirements into "economic batch quantities" and to adjust the programme for unbalanced buffer stocks.

The required buffer at the end of the current period, which is determined by the manufacturing lead time, is calculated as the gross requirements extended by the number of weeks of planned buffer. The latter is a prime experimental parameter.

The net result at the end of the period is derived by comparing total requirements (planned buffer, gross demand, allocated stock) with the total available stock (physical stock, work in progress, quantity in allocation).

The nett quantity, if positive, will cause requirements to be deleted until the excess has been consumed. A negative nett quantity will cause a batch to be planned in the current period.

For the period beyond the modified zone, batches are planned according to the frequency and batch size contained in file IFILE.

The resultant file, SUBFILE, is used to plan the manufacture of sub-assembly batches, initiated by subroutine ALLOCATE.
Summarised Flow Chart - Subroutine Subprog
SUBPROG

SUBROUTINE SUBPROG(REQFILE, WEEK, IFILE, IPRINT, SUBFILE, AFILE, ALLQ)
REAL AFILE(25, 3)
INTEGER REQFILE(25, 52), IFILE(25, 23), WEEK, SUBFILE(12, 26), BUFFER, DEM
1AND, ALLQ(12, 50, 6), ALLQTY
IF(IPRINT .GT. 4) GO TO 1
WRITE(2, 205)
205 FORMAT(1HO, 6X, 'SUB-ASSEMBLY PROGRAMMES')
DO 50 I = 1, 12
IF(IFILE(I, 11) .EQ. 2 .OR. IFILE(I, 11) .EQ. 3) GO TO 5
GO TO 50
5 IF(IFILE(I, 15) .NE. 1) GO TO 2
DO 6 J = 1, 26
6 SUBFILE(I, J) = 0
GO TO 110

FIND QUANTITY IN ALLOCATION

2 ALLQTY = 0
DO 3 J = 1, 750
3 ALLQTY = ALLQTY + ALLQ(I, J)

CALCULATE BUFFER AT PERIOD END

BUFFER = 0
IF(IPRINT .GT. 1) GO TO 7
WRITE(2, 200) AFILE(I, 1)
200 FORMAT(1HO, 6X, 'FILE =', 18)
7 K = IFILE(I, 2) + WEEK + IFILE(I, 12) + 1
IF(IPRINT .GT. 1) GO TO 9
WRITE(2, 201) K
201 FORMAT(1HO, 6X, 'K =', 14)
9 IF(K .GT. S2) GO TO 50
L = WEEK + IFILE(I, 12) + 2
DO 10 J = L, K
10 BUFFER = BUFFER + REQFILE(I, J)

CALCULATE DEMAND BEFORE NEXT BATCH CAN BE ALLOCATED AND MADE

15 DEMAND = 0
M = WEEK + IFILE(I, 12) + 1
IF(M .GT. 52) GO TO 50
DO 20 L = WEEK, M
20 DEMAND = DEMAND + REQFILE(I, L)
IF(IPRINT .GT. 1) GO TO 25
WRITE(2, 203) DEMAND, BUFFER, ALLQTY
203 FORMAT(1HO, 6X, 'DEMAND =', 16, 'BUFFER =', 16, 'ALLQTY =', 16)

CALCULATE BATCH SIZE AND TIMING

25 L = WEEK + 12
IF(L .LE. 26) GO TO 30
L = 26
30 DO 40 J = WEEK, L
40 SUBFILE(I, J) = 0
NETQTY = IFILE(I, 16) - IFILE(I, 17) - BUFFER + IFILE(I, 18) - DEMAND + ALLQTY
IF(IPRINT .GT. 1) GO TO 45
WRITE(2, 204) NETQTY
204 FORMAT(1HO, 6X, 'NETQTY =', 16)
N=IFILE(I,22)
  J=WEEK
  IF(NETQTY.LT.0)GO TO 60
  DO 70 J=WEEK;L
  IF(REQFILE(I,J).GE.NETQTY)GO TO 60
  NETQTY=NETQTY-REQFILE(I,J)
  SUBFILE(I,J)=0
70 CONTINUE
  GO TO 110
60 SUBFILE(I,J)=IFILE(I,J)
  K=J+N
  IF(IPRINT.GT.1)GO TO 95
90 WRITE(2,500)(K,L,N
500 FORMAT(1HO,6X,K=I14,L=I14,N=I14)
95 DO 100 J=K,L;N
100 SUBFILE(I,J)=IFILE(I,J)
110 IF(IPRINT.GT.4)GO TO 50
  WRITE(2,600)(AFILE(I,J),SUBFILE(I,J);J=1,20)
600 FORMA(IHO,6X,A8,2X,20)
50 CONTINUE
  RETURN
END
An essential ingredient in the design of the model was to make communication both to and from the model as simple as possible. The facility has been provided to define part number data in alpha-numeric code rather than restricting the modeller to numeric only coding, thus allowing the use of familiar code references and simplifying the diagnostic routine interpretation, where code numbers could easily be confused with parameters and variable data. The only constraint in code numbering is that the field length is restricted to eight characters.

The subroutine accepts a real word from the calling programme, and compares each of the eight characters with the type code contained within AFILE by use of library subroutine COMP8. If a successful match is made, the logical position of the AFILE entry is returned as the equivalent numeric parameters defining the code number.

Summarised Flow Chart - Subroutine Translate 3
SUBROUTINE TRANSLATE (AFILE, TYPE, NUMBER, *)
REAL AFILE(25,3), TYPE
DO 10 I=1,25
CALL COMP8(TYPE, AFILE(I,1), ITEST)
GO TO (30, 10) ITEST
10 CONTINUE
WRITE(2, 20) TYPE
20 FORMAT(1HO.6X,A8,'IS AN INVALID PART NUMBER')
RETURN 1
30 NUMBER = I
RETURN
END

TRANSLATE 3
Subroutine TRENDI introduces a trend factor into the product mix sampling histogram. Thus, the behaviour of the model under conditions of sales growth and decay may be observed. Since the subroutine modifies only the probability that a particular product will be selected and not its quantity, the overall business volume is not influenced.

The product mix parameters available to the customer order generator, subroutine ORDERS, are held in the integer file TABLE 6. The initial values of product mix are derived from TABLE 6 and then translated into the real file PRODMIX for subsequent processing.

Trend parameters may be maintained independently for each product and are located in file TREND. The trend values represent the per unit increase (or decrease for negative trend values) in product mix value per period. Subroutine TRENDI is activated in the processing of subroutine QTREVENTS, thus each period represents a three month interval.

The direct application of trend modifiers to the file PRODMIX will normally result in a difference between the sum of the product components and the original scale of 0 - 100. Each component is therefore modified back to this scale and the cumulative mix results loaded into file TABLE 6.
SUMMARISED FLOW CHART - SUBROUTINE TRENID

SUBROUTINE TRENID(TABLE6, PRODMIX, TREND, IPRINT)
REAL PRODMIX(7), TREND(7)
INTEGER TABLE6(7, 2)
SUM = 0.0
DO 50 I = 1, 7
   PRODMIX(I) = PRODMIX(I) * (1 + TREND(I))
   SUM = SUM + PRODMIX(I)
50 CONTINUE
IF(IPRINT .GT. 3) GO TO 65
WRITE(*, 2.60) (PRODMIX(I), I = 1, 7)
60 FORMAT(1HO, 6X, \"PRODMIX WITH TREND\"/10X, 7F7.2)
IF(IPRINT .GT. 1) GO TO 65
WRITE(*, 2.63) SUM
63 FORMAT(1HO, 6X, \"SUM = \"/10X, 7F7.2)
65 Z = 0
DO 70 I = 1, 7
   Z = Z + (PRODMIX(I) * 100.0 / SUM)
70 TABLE6(1:2) = INT(Z)
IF(IPRINT .GT. 3) GO TO 90
WRITE(*, 2.80) (TABLE6(1:2), I = 1, 7), (TABLE6(1:2), I = 1, 7)
80 FORMAT(1HO, 6X, \"REVISED TABLE\"/10X, 7F16/10X, 7F16//)
90 RETURN
END
Subroutine WEEKEVENT sequences the weekly activities and links the data transfer between segments. The sequence of each segment within the subroutine is critical to the system performance and integrity, since it must follow, as closely as possible, the real world.

The subroutine simulates the nett effect of a week's worth of activities. No facility is provided to simulate a shorter time interval since such an approach would substantially reduce the computational efficiency and would provide little benefit. The main disadvantage is that no more than one activity may take place on a specific order in one week.

The first activity each period is the generation of a new batch of customer orders, which are filed in the orders pipeline. File PIPELINE may optionally be printed for diagnostic purposes.

Orders are extracted from the pipeline and loaded against the delivery plan in subroutine LOAD.

Components which are short to allocation are expedited in subroutine EXPEDITE and material receipts are simulated by subroutine ARRIVAL.

Customer orders which are still in unplanned status are selected for allocation against free equipment stock or for make to order in subroutine ALLOCATE. Subroutine ALLOCATE will subsequently cause components to be reserved and the order to be entered into the allocation queue in subroutine STKALL, together with stock replenishment orders as required to complete a batch. Orders selected for allocation against stock will be moved into "ex-stock" status for subsequent transfer into the test queue, using subroutine FESALL, as the due date approaches.

Orders in the allocation queue are moved into assembly work-in-progress by subroutine ISSWIP when all of the components are available, making allowance for the component picking lead time.
Orders which have completed the assembly process are moved out of the assembly queue in subroutine OFFLINE. Stock orders will update finished equipment stock in preparation for allocation in subroutine FESALL. Customer orders are moved directly to the test work-in-progress queue.

Subroutine FESALL selects orders in "ex-stock" status and compares the quantity required with the physical stock of finished equipment. Orders are subsequently moved into the test work-in-progress queue if stock exists.

Subroutine DESPATCH selects orders from the test work-in-progress queue according to the priority rules selected and transfers the products to "despatch" status. Items transferred are analysed by subroutine DELPERF which will:

a) determine whether the order, or part order may be despatched, according to the part shipment rules;

b) update the delivery performance statistics.

Items which do not satisfy the part shipment rules remain in "despatch" status until the part shipment rules are satisfied.

The status of stock in the item master file may optionally be displayed at the end of each weekly cycle.
Summarised Flow Chart - Subroutine Wekeevent
SUBROUTINE WEEKEVENT(AFILF, IDLE, TABLE3, TABLE5, TABLE6, TABLE7, TABLE8, AVGE9, CVARN9, AVGE10, CVARN10, AVGE12, CVARN12, C3VAR10, ORDCOUNT, PIPELINE, DEPLAN, OFCFMIN, WEEK, TOTAL, ORDBK, OPFILE, ORDQHIST, INPUT, EXFILE, SCHEDULE, REQFILE, ALLQ, LINEQ, TESTQ, SUBFILE, EQUPERF, ORDPERF, NETPLAN, REFWEEK, IPRINT, ICOUNT, ISHORT, CAPACITY; LEADTM, ISORT, SUBQ, DIAGFILE, MODE, RUNTIME)

REAL AFILF(25, 3)
INTEGER [FILE(25, 2), TABLE3(7, 2), TABLE5(7, 2), TABLE6(7, 2), TABLE7(7, 2), TABLE8(2, 2), TABLE11(2, 2), ORDCOUNT(20, 6), PIPELINE(20, 5), OFCFMIN(5, 5), WEEK(5, 5), TOTAL(5, 5), ORDBK(5, 5), OPFILE(125, 5, 5), ORDQHIST(50, 6), INPUT(25, 52), EXFILE(25, 52), SCHEDULE(25, 13), REQFILE(25, 52), STKORD(20), ALLQ(12, 50, 6), LINEQ(5, 50, 5), TESTQ(5, 50, 5), SUBFILE(12, 26), EQUPERF(10, 30), ORDPERF(2, 30), NETPLAN(5, 52), REFWEEK(10, 2), IPRINT(1), ICOUNT(25), ISHORT(25)

0 CONTINUE
1 DO 5 I=1, 10
2 IF (DIAGFILE(I, 1) .GT. (REFWEEK+WEEK)) GO TO 7
5 CONTINUE
10 FORMAT(IPRINT.DIAGFILE(I, 1) GO TO 30
WRITE(2, 10)
20 RETURN
END
The timebase for the prime plan recalculation is three-monthly (quarterly) periods. This time period is too coarse for detail factory scheduling, for which purpose the plan is expanded into weekly "buckets" or time periods.

Subroutine WEEKLYPLAN performs this task at two levels, following a common logic.

- the off-line plan, OFP, is translated into a weekly plan in REQFILE
- the production to be allocated, PA, is translated into a delivery plan, DELPLAN.

In practice, the computer logic within the Purchasing and Scheduling sub-system will cause the quarterly data to be divided by thirteen and this provides an equivalent weekly plan.

The facility is available to modify the suggested plan by a manual override where it is desirable to reflect significant changes in planned activity through the period, or to more clearly define batch produced equipments.

The subroutine WEEKLYPLAN combines the two requirements into one operation.

Input to the subroutine is the quarterly plan derived from subroutine QPLAN. Based on the magnitude of the quarterly requirements, the decision is then made on the frequency of manufacture according to the table below.

<table>
<thead>
<tr>
<th>QUARTERLY REQUIREMENT (x)</th>
<th>FREQUENCY OF MANUFACTURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>x ≤ 50</td>
<td>✓</td>
</tr>
<tr>
<td>50 &lt; x ≤ 150</td>
<td>✓</td>
</tr>
<tr>
<td>150 ≤ x</td>
<td>✓</td>
</tr>
</tbody>
</table>
The programme will divide the quarterly requirement by the frequency of manufacture and schedule the resultant batch to the first week of the sub-period.

Thus, a requirement of 40 equipments will be scheduled in its entirety for the first week of the quarter; and a requirement of 120 equipments will be split into three equal batches of 40 (one a month) scheduled for the first week of each month. It should be noted that, since the quarter does not conveniently divide by 3, the months are defined as commencing in weeks 1, 6 and 10 of each quarter.

The resultant weekly plan for the equipment is held in an array which defines the forward requirements over a 52 week planning horizon.

Two assumptions have been made to avoid over-complication of the logic, neither of which will have a significant effect on the operation of the model.

a) No account is taken of the actual number of working days in each quarter, thus each quarter is assumed to comprise 13 equal weeks.

b) No attempt is made to even out the loading of batched products. In practice, such a policy would cause an uneven capacity profile throughout the quarter, but this is not considered to be a major consideration in the small sub-set of products involved in the experiments.
Summarised Flow Chart - Subroutine Weeklyplan
SUBROUTINE WEEKLYPLAN(OFP, REQFILE, IFILE, PA, DELPLAN, IPRINT, AFILE)
REAL AFILE(25,3)
INTEGER OFP(10,7), REQFILE(25,52), IFILE(25,23), PA(10,10), DELPLAN(151,52)
DO 60 K = 1, 5
IF(IFILE(K,11).NE.1) GO TO 60
M = 1
DO 30 I = 2, 5
DO 20 J = M, N
IF(OFP(K, I).GT.50) GO TO 10
REQFILE(K, J) = OFP(K, I)
DELPLAN(K, J + 2) = PA(K, I)
GO TO 25
10 IF(OFP(K, I).GT.150) GO TO 18
DO 15 L = J, J + 9, 4
DELPLAN(K, L + 2) = PA(K, I)/3.0
15 REQFILE(K, L) = OFP(K, I)/3.0
GO TO 25
18 DELPLAN(K, J) = PA(K, I)/3.0
20 REQFILE(K, J) = OFP(K, I)/3.0
25 N = M + 13
N = N + 13
30 CONTINUE
IF(IPRINT.GT.3) GO TO 60
WRITE(2, 50) AFILE(K, 1), (REQFILE(K, J), J = 1, 52), (DELPLAN(K, J), J = 1, 52)
50 FORMAT(1HO//6X,'OFF-LINE PLAN FOR PRODUCT REF. ', A8//6X,26I4//6X,18614//6X,'DELIVERY PLAN',//6X,26I4//6X,26I4)
60 CONTINUE
RETURN
END
Subroutine WEEKPRINT is one of the prime output segments, and is used to calculate and display statistics relating to service level and delivery performance.

The subroutine will be activated either weekly or quarterly, dependent upon the diagnostic level selected. The data files and status variables analysed by WEEKPRINT are initialised by subroutine QTRRESET, following the quarterly activation.

Component service level is defined as the number of times an item can be supplied from stock as a ratio of the number of times requested. Service level statistics are maintained and presented both by individual item number and as a total for all items, to permit further detailed analysis against the item usage (i.e. common part or specific to an product type).

Delivery performance statistics are maintained in subroutine DELPERF at three levels; equipment, item and order. Data is contained within frequency histograms which relate the delivery date achieved with the number of instances. Delivery performance is calculated and presented in two alternative modes; proportion delivered on or before the due date, or the average lateness. The raw histogram data is available for further analysis if required.

Equipment performance is presented by product type and in total for all equipments; item performance is by product type only. Order performance is presented separately for "part shipment" and "no part shipment" orders.

A further measure of customer satisfaction, the quoted delivery lead time, is also maintained in histogram form for further analysis if required.
CALCULATE SUMMARY EQUIP. PERFORMANCE

CALCULATE SERVICE LEVELS

WRITE SERVICE LEVELS

CALCULATE EQUIPMENT PERFORMANCE AND PRINT

CALCULATE TOTAL PERFORMANCE AND PRINT

CALCULATE ITEM PERFORMANCE AND PRINT

SELECT PART SHIP THEN NO PART SHIP

CALCULATE ORDER PERFORMANCE AND PRINT

CALCULATE LEAD TIMES AND PRINT

RETURN

Summarised Flow Chart - Subroutine Weekprint
SUBROUTINE WEEKPRINT (ICOUNT,ISHORT,EQUPERF,ITEMPERF,ORDPERF,IFILE, 
AFILE,REFWEEK,WEAK,LEADTM)
REAL AFILF(25,3) INTEGER EQUPERF(10,30),ITEMPERF(10,30),ORDPERF(2,30),KK(30),IFILE(125,23),REFWEEK,WEAK,LEADTM(5,52),TOTPERF(30),ICOUNT(25),ISHORT(25)
I=REFWEEK+WEAK
WRITE(2,10)
10 FORMAT(1HO,6X,'* * * SUMMARY FOR WEEK', I6, ' * * *')
DO 5 J=1,30
TOTPERF(J)=0
DO 5 I=1,10
5 CONTINUE
ICTOT=0
ISTOT=0
WRITE(2,90)
90 FORMAT(1HO,2X,'SERVICE LEVEL ACHIEVED'/11X,'ITEM NUMBER',6X, 
 ITEMS REQUIRED',6X,'ITEMS SHORT',6X,'SERVICE LEVEL')
DO 50 I=1,25
IF(IFILE(I,11).NE.4)GO TO 50
ICTOT=ICTOT+ICOUNT(I)
ISTOT=ISTOT+ISHORT(I)
SERV=100*FLOAT(ICTOT-ISHORT(I))/FLOAT(ICOUNT(I))
WRITE(2,40)AFILF(I,1),ICOUNT(I),ISHORT(I),SERV
40 FORMAT(1HO,11X,'A8';12X,16,12X;16,12X,F6.2)
50 CONTINUE
SERV=100*FLOAT(ICTOT-ISTOT)/FLOAT(ICTOT)
WRITE(2,60)ICTOT,ISTOT,SERV
60 FORMAT(1HO,11X,'TOTAL'/15X,16,12X,16,12X,F6.2)
DO 130 J=1,29
130 KK(J+1)=KK(J)+1
WRITE(2,200)(KK(J),J=1,30)
200 FORMAT(1HO//6X,'DELIVERY PERFORMANCE'/304//6X,'EQUIPMENT')
DO 180 I=1,10
IF(IFILE(I,11).NE.1)GO TO 180
SUM1=0.0
SUM2=0.0
SUM3=0.0
SUM4=0.0
SUM5=0.0
DO 140 J=1,30
140 SUM1=SUM1+EQUPERF(I,J)
IF(SUM1.EQ.0.0)GO TO 175
DO 150 J=1,30
150 SUM2=SUM2+EQUPERF(I,J)
SUM2=SUM2*100.0/SUM1
DO 160 J=1,30
160 SUM3=SUM3+EQUPERF(I,J)
SUM3=SUM3*100.0/SUM1
DO 170 J=1,30
170 SUM4=SUM4+EQUPERF(I,J)
SUM4=SUM4*100.0/SUM1
DO 173 J=1,30
173 SUM5=SUM5+EQUPERF(I,J)*J
SUM5=21.0-SUM5/SUM1
175 WRITE(2,500) AFILE(I,1)
500 FORMAT(1HO,6X,48)
   WRITE(2,600) SUM2, SUM3, SUM4, SUM5
600 FORMAT(1HO,6X,30X,1ON OR BEFORE=1, F6.2, 1% WITHIN 3 WEEKS=1, F6.2, 1% WITHIN 6 WEEKS=1, F6.2, 1% LATENESS=1, F6.2)
   WRITE(2,700) (EQUIPERF(I,J), J=1,30)
700 FORMAT(1HO,3014/)
180 CONTINUE
   SUM1=0.0
   SUM2=0.0
   SUM3=0.0
   SUM4=0.0
   SUM5=0.0
   DO 185 J=1,30
185 SUM1=SUM1+TOTPERF(J)
   IF(SUM1.EQ.0.0) GO TO 195
   DO 187 J=21,30
187 SUM2=SUM2+TOTPERF(J)
   SUM2=SUM2*100.0/SUM1
   DO 189 J=15,30
189 SUM3=SUM3+TOTPERF(J)
   SUM3=SUM3*100.0/SUM1
   DO 191 J=1,30
191 SUM4=SUM4+TOTPERF(J)
   SUM4=SUM4*100.0/SUM1
   DO 193 J=1,30
193 SUM5=SUM5+TOTPERF(J)*J
   SUM5=SUM5/SUM1
195 WRITE(2,197)
197 FORMAT(1HO,6X,TOTAL1)
   WRITE(2,600) SUM2, SUM3, SUM4, SUM5
   WRITE(2,700) (TOTPERF(J), J=1,30)
   WRITE(2,800)
800 FORMAT(1HO,6X,ITEM1)
   DO 280 I=1,10
      IF(IFILE(I,11).NE.1) GO TO 280
      SUM1=0.0
      SUM2=0.0
      SUM3=0.0
      SUM4=0.0
      SUM5=0.0
      DO 240 J=1,30
240 SUM1=SUM1+ITEMPERF(I,J)
      IF(SUM1.EQ.0.0) GO TO 275
      DO 250 J=21,30
250 SUM2=SUM2+ITEMPERF(I,J)
      SUM2=SUM2*100.0/SUM1
      DO 260 J=15,30
260 SUM3=SUM3+ITEMPERF(I,J)
      SUM3=SUM3*100.0/SUM1
      DO 270 J=1,30
270 SUM4=SUM4+ITEMPERF(I,J)
      SUM4=SUM4*100.0/SUM1
      DO 273 J=1,30
273 SUM5=SUM5+ITEMPERF(I,J)*J
      SUM5=SUM5/SUM1
275 WRITE(2,500) AFILE(I,1)
      WRITE(2,600) SUM2, SUM3, SUM4, SUM5
      WRITE(2,700) (ITEMPERF(I,J), J=1,30)
280 CONTINUE
WRITE(2,900)
900 FORMAT(1H0,6X,1ORDERS1)
DO 390 I=1,2
SUM1=0.0
SUM2=0.0
SUM3=0.0
SUM4=0.0
SUM5=0.0
DO 340 J=1,30
340 SUM1=SUM1+ORDPERF(I,J)
IF(SUM1.EQ.0.0)GO TO 371
DO 350 J=21,30
350 SUM2=SUM2+ORDPERF(I,J)
SUM2=SUM2*100.0/SUM1
DO 360 J=18,30
360 SUM3=SUM3+ORDPERF(I,J)
SUM3=SUM3*100.0/SUM1
DO 370 J=15,30
370 SUM4=SUM4+ORDPERF(I,J)
SUM4=SUM4*100.0/SUM1
DO 373 J=1,30
373 SUM5=SUM5+ORDPERF(I,J)*J
SUM5=21.0-SUM5/SUM1
371 GO TO(372,376),1
372 WRITE(2,374)
374 FORMAT(1H0,6X,1PART SHIP1)
GO TO 380
376 WRITE(2,378)
378 FORMAT(1H0,6X,1NO PART SHIP1)
380 WRITE(2,600)SUM2, SUM3, SUM4, SUM5
WRITE(2,700)(ORDPERF(I,J), J=1,30)
390 CONTINUE
WRITE(2,450)(J,J=1,52)
450 FORMAT(1H0/6X,1LEAD TIME ACHIEVED1/2(/5X,26I4))
DO 490 I=1,5
IF(IFILE(I,11).NE.1)GO TO 490
WRITE(2,470)(AFILE(I,11), LEADTH(I,J), J=1,52)
470 FORMAT(1H0,6X,AB/2(/5X,26I4))
490 CONTINUE
RETURN
END
FINISH
****
DELIVERY PERFORMANCE

<table>
<thead>
<tr>
<th>EQUIPMENT</th>
<th>ON OR BEFORE</th>
<th>0.00% WITHIN 3 WEEKS</th>
<th>100.00% WITHIN 6 WEEKS</th>
<th>100.00% LATENESS</th>
<th>0.00X WITHIN 6 WEEKS</th>
<th>100.00X LATENESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF4AM 01</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF4AM 02</td>
<td>12.50%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF4AM 03</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M16PU</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>3.40%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF4AM 04</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF4AM 02</td>
<td>12.50%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF4AM 03</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>3.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M16PU</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>2.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORDERS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PART SHIP</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>1.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NO PART SHIP</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00%</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sample Delivery Performance Report
<table>
<thead>
<tr>
<th>ITEM NUMBER</th>
<th>ITEMS REQUIRED</th>
<th>ITEMS SHORT</th>
<th>SERVICE LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>RU69873</td>
<td>11</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>RU10034</td>
<td>10</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>BT69863</td>
<td>3</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>PP42906</td>
<td>5</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>PN50006</td>
<td>3</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>PN10638</td>
<td>1</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>PN10639</td>
<td>6</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>PN56043</td>
<td>2</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>PN69746</td>
<td>10</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>ET12345</td>
<td>3</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>FU10000</td>
<td>2</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>FV25000</td>
<td>3</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>FS66000</td>
<td>2</td>
<td>0</td>
<td>100.00</td>
</tr>
<tr>
<td>TOTAL</td>
<td>61</td>
<td>0</td>
<td>100.00</td>
</tr>
</tbody>
</table>

Sample Component Service Level Report
APPENDIX C

FILE DESCRIPTIONS

Each file required by the simulation model is defined for ease of maintenance and analysis. The definition comprises the following information.

File Description: Briefly describes the content and use of the file.

File Content: The name of the file as used in the simulation program, and the definitions of the use of each array subscript.

File Size: The dimensions of the file.

File Type: The file will contain either integer or real (floating point) values.

Program Segments: The subroutines which contain the file. This information is required if the file sizes require modification.

Special Notes: Any important information concerning the structure, content or use of the file.
File Name:        AFILE

File Description: Item Master File (Part)

Contains descriptive (text) part of the Item Master Record, used mainly for ease of communication between the modeller and the system.

File content:     AFILE (a, b)
                   a = file address
                   b = 1 item number (up to 8 alpha-numeric characters)
                   2) item description (up to 16 alpha-
                   3) numeric characters)

File size:        AFILE (25,3) = 75

File type:        Real

Program Segments: STOCKMODEL, ALLOCATE, AQPRINT, ARRIVAL, DESPATCH, EBQ, EXPLODE, FESALL, FORECAST, ISSWIP, LOAD, LOADPLAN, LQPRINT, MONTHEVENT, OFFLINE, OPFCREATE, OPFPRT, ORDBKPRT, PLNPRINT, QPLANPRINT, QEVENTS, QRTRESET, PNMFCREATE, PNMFPRINT, RECEIVE, SCHED, SQPRINT, STKALL, SUBPROG, TQPRINT, TRANSLATE 3, WEEKEVENT, WEEKLYPLAN, WEEKPRINT

Special Notes:    File contains static data which is created prior to program execution and is not modified in any way during the simulation run.

The internal system logic will refer to items by a numeric key, which is equivalent to the address of the item in AFILE.
File Name: AFSMAX

File Description: Maximum authorised stock

Contains the maximum permissible finished equipment stock level per product.

File Content: AFSMAX (a)

a = product key/quantity

File Size: AFSMAX (10) = 10

File Type: Integer

Program Segements: QPLAN, QPLANPRINT

Special Notes: Used to convey data between QPLAN and QPLANPRINT for presentation of parameters.
File Name: ALLQ

File Description: Allocation queue

Contains a queue of requirements which have been allocated to stock pending release of parts to work in progress.

File content: ALLQ (a, b, c)

a = Product key
b = Position in file
c = 1 .... Order number
2 .... Item number
3 .... Due date
4 .... Quantity
5 .... Timer
6 .... Shop order number

File Size: ALLQ (12, 50, 6) = 3600

File Type: Integer

Program Segments: STOCKMODEL, ALLOCATE, AQPRINT, ISSWIP, MONTHEVENT, OPFCREATE, QTREVENTS, WEEKEVENT

Special Notes: File is organised into a number of separate queues, one for each assembly number, each containing a sequence of material issue requirements. The file will be accessed sequentially, thus permitting a choice of priority rules; sorted by due date (utilising a sort pre-processor) or "first in first out". After processing, the file is always consolidated, and new entries are added to the tail of the file.
File Name: COL

File Description: Current order load

Contains the nett order load at the commencement of a quarterly period.

File Content: COL (a, b)

a = product key
b = 1 .... quantity overdue
     2-7 .. quantity due in periods 1-6

File Size: COL (10, 7) = 70

File Type: Integer

Program Segments: STOCKMODEL, QPLAN, QPLANPRINT, QTREVENTS

Special Notes: None
File Name: DELPLAN

File Description: Delivery plan

Plan of proposed deliveries from the factory to the commercial warehouse, used for order promising and relief of test work in progress.

File Content: DELPLAN (a, b)

a = Product key
b = 1-52 .... Plan quantity per period

File Size: DELPLAN (5, 52) = 260

File Type: Integer

Program Segments: STOCKMODEL, DESPATCH, LOAD, LOADPLAN, MONTHEVENT, QTREVENTS, WEEKEVENT, WEEKLYPLAN

Special Notes: File is organised into weekly buckets, each containing the planned delivery quantity for the period. The file is reset each quarter by replacing the planned quantities such that the first bucket is the current period.
File Name: DIAGFILE

File Description: Diagnostic Selector

Contains instructions which set the level of diagnostic output for each simulation run.

File Content: DIAGFILE (a, b)

\[\begin{align*}
a &= \text{position in file} \\
b &= 1 \quad \text{... week number (actual)} \\
    &= 2 \quad \text{... diagnostic level}
\end{align*}\]

File size: DIAGFILE (10, 2) = 20

File Type: Integer

Program Segments: STOCKMODEL, ALLOCATE, DESPATCH, FESALL, ISSWIP, OFFLINE, WEEKEVENT

Special Notes: The use of DIAGFILE permits the resetting of the diagnostic level at any week number during the simulation run, thus permitting a detailed analysis of a particular time frame during the run.

The file is organised as follows:

\[
\begin{array}{cccccccccc}
\text{Week No} & 1 & 14 & 20 & 22 & 60 & 999 & 0 & 0 & 0 & 0 \\
\text{Diag level} & 1 & 5 & 2 & 4 & 6 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

The above example will set the level to '1' for the initial conditions and then reset to '5' for periods 14 - 19 inclusive. The level will then revert to '1' for periods 20/21, reset to '4' for periods 22 to 59 and '6' to the end of the run.
File Name: EQUPERF

File Description: Equipment performance

Contains statistics relating to the delivery performance achievement at equipment level.

File Content: EQUPERF (a, b)

a = product key
b = 1 - 30 .... frequency histogram over the range -20 to +9.

File Size: EQUPERF (10, 30) = 300

File Type: Integer

Program Segments: STOCKMODEL, DELPERF, DESPATCH, WEEKEVENT, WEEKPRINT

Special Notes: File is organised as a frequency histogram, for each product, containing 30 intervals. Each product batch despatched augments the appropriate interval by the batch quantity. Delays over the range -20 to +9 weeks are converted to the file range 1 to 30 as data is captured, and re-converted to represent the correct delay factor for subsequent analysis.
File Name: EXFILE

File Description: Expedite tag

Contains a counter in each week that an item has been subject to expediting action.

File Content: EXFILE (a, b)

a = product key
b = 1-52 .... weekly buckets containing expedite action counter

File Size: EXFILE (25, 52) = 1300

File Type: Integer

Program Segments: STOCKMODEL, EXPEDITE, WEEKEVENT

Special Notes: File is reset to zero each quarter. The appropriate weekly bucket is incremented each time that a receipt batch is subject to expediting action, thus providing a measure of the expediting effort required.
File Name: FES

File Description: Finished Equipment Stock

Contains the planned level of finished equipment stock per quarterly period.

File Content: FES (a, b)

a = product key
b = 1 .... opening stock (actual)
    2-7 .. planned closing stock for periods 1 to 6

File Size: FES (10, 7) = 70

File Type: Integer

Program Segments: QPLAN, QPLANPRINT

Special Notes: None
File Name: HOLD (also IHOLD)

File Description: Temporary workfile

Workfile used to store the item number and quantity of lower level assemblies during material explosion sequences.

File Content: HOLD (a, b)

a = position in file
b = 1 .... Item number
    2 .... Quantity

File Size: HOLD (5, 2) = 10

File Type: Integer

Program Segments: ISSWIP, STKALL

Special Notes: File is only required where the lower level part is classified as issue code (1), which means that the assembly should not be planned as a stocked part, but should be broken down to its constituent parts, i.e. a phantom assembly.
File Name: IFILE

File Description: Item Master File (part)

Contains data related to the item number.

File Content: IFILE (a, b)

\[
a = \text{product key} \\
b = 1 \quad \text{Location of 1st component} \\
2 \quad \text{Quantity of 1st component} \\
3 \quad \text{Location of 2nd component} \\
4 \quad \text{Quantity of 2nd component} \\
5 \quad \text{Location of 3rd component} \\
6 \quad \text{Quantity of 3rd component} \\
7 \quad \text{Location of 4th component} \\
8 \quad \text{Quantity of 4th component} \\
9 \quad \text{Location of 5th component} \\
10 \quad \text{Quantity of 5th component} \\
11 \quad \text{Low Level code (1 - 4)} \\
12 \quad \text{Lead Time} \\
13 \quad \text{Material Cost} \\
14 \quad \text{Labour Cost} \quad \text{Cost Data} \\
15 \quad \text{Issue Code} (1 = \text{normal} \\
\quad \quad 2 = \text{do not breakdown}) \\
16 \quad \text{Physical Stock} \\
17 \quad \text{Allocated Stock} \quad \text{Stock} \\
18 \quad \text{Work in progress (assembly)} \quad \text{Status} \\
19 \quad \text{Work in progress (test)} \\
20 \quad \text{Buffer in Weeks} \\
21 \quad \text{Economic Batch Quantity} \quad \text{Planning} \\
22 \quad \text{Cycle Time between Batches} \quad \text{Parameters} \\
23 \quad \text{Maximum Authorised Stock}
\]

FileSize: IFILE (25, 23)

FileType: Integer
Program Segments: STOCKMODEL, ALLOCATE, ARRIVAL, DESPATCH, EBQ, EXPEDITED, FESALL, FORECAST, INITOOS, ISSWIP, LOAD, LOADPLAN, OFFLINE, OFFCREATE, ORDBKPRIT, ORDERS, QPLAN, QPLANPRINT, QCREVENTS, QRESET, PMCREATE, PMPRINT, RECEIVE, SCHED, STKALL, STOCKVAL, WEEKEVENT, WEEKLYPLAN, WEEKPRINT

Special Notes: File contains a number of segments relating to the item. Static data includes the product structure relationships, cost data and planning parameters. The stock status segment contains dynamic data relating to stores and work in progress stocks.
File Name: INPUT

File Description: Weekly material input.

Contains a schedule of the actual material receipts.

File Content: INPUT (a, b)
   a = product key
   b = 1-52 .... Material input per period

File Size: INPUT (25, 52) = 1300

File Type: Integer

Program Segments: STOCKMODEL, ARRIVAL, EXPEDITE, QTREVENTS,
   QTRRESET, RECEIVE, WEEKEVENT.

Special Notes: The file is reset each quarter by advancing each
data element by thirteen weeks.

Data in INPUT is periodically created within
QTREVENTS by subroutine RECEIVE. The contents of
INPUT may be modified in one of two ways;

- batches may be advanced in time as a result
  of expediting action (subroutine EXPEDITE).
- reject quantities (subroutine ARRIVAL) will
  be re-scheduled to a later period.
File Name: IONP

File Description: Initial On-line plan

Contains the on-line (material) plan prior to the quarterly plan regeneration.

File Content: IONP(a, b)

a = product key
b = 1-6 .. on-line plan for periods -1 to 5 inclusive.
7 .... not used

File Size: IONP(10, 7) = 70

File Type: Integer

Program Segments: QPLAN, QPLANPRINT, QTREVENTS

Special Notes: None
File Name: ITEMPERF

File Description: Item performance.

Contains statistics relating to the delivery performance achievements at order line level.

File Content: ITEMPERF (a, b)

a = product key
b = 1-30 .... frequency histogram over the range -20 to +9

File Size: ITEMPERF (19,30) = 300

File Type: Integer

Program Segments: STOCKMODEL, DELPERF, DESPATCH, WEEKEVENT, WEEKPRINT.

Special Notes: File is organised as a frequency histogram for each product, containing 30 intervals. Each completed item line will augment the appropriate interval by one.
File Name: LEADTM

File Description: Lead time

Contains data for compiling a frequency histogram of quoted delivery lead time for each product.

File Content: LEADTM (a, b)

a = product key
b = 1-52 .... frequency of achieving each value

File Size: LEADTM (5, 52) = 260

File Type: Integer

Program Segments: STOCKMODEL, LOAD, QTREVENTS, QTRRESET, WEEKEVENTS, WEEKPRINT.

Special Notes: As each order is loaded, the delivery lead time bucket for the appropriate product is augmented by the quantity loaded.
File Name: LINEQ

File Description: Assembly Work in Progress

Contains a queue of customer and stock orders for each product type at the main assembly location.

File Content: LINEQ (a, b, c)

a = Product key
b = Position in file
c = 1 .... order number
    2 .... item number
    3 .... due date
    4 .... quantity
    5 .... timer

File Size: LINEQ (5, 50, 5) = 1250

File Type: Integer

Program Segments: STOCKMODEL, WEEKEVENT, ALLOCATE, ISSWIP, OFFLINE

Special Notes: File is organised into a separate queue for each product type (sub-assemblies are located in SUBQ).

New entries to the file are always placed at the end of the queue. Two priority rules may be utilised; "first in first out", or by due date - in which case a sort routine is activated. After processing the file is consolidated by removing fully processed records.
File Name: NETPLAN

File Description: Order loading plan.
Contains the order loading plan for each product type.

File Content: NETPLAN (a, b)

\[ a = \text{product key} \]
\[ b = 1-52 \ldots \text{Weekly order loading plan} \]

File Size: NETPLAN (5, 52) = 260

File Type: Integer

Program Segments: STOCKMODEL, LOADPLAN, MONTHEVENT, QTREVENT,
QTRRESET, WEEKEVENT.

Special Notes: The gross delivery plan is contained in file DELPLAN. The net delivery plan is derived from the gross delivery plan with modifications reflecting the level of overdue products and the difference between the planned and actual stock levels. The resultant plan is loaded into NETPLAN and used as the basis for order promising.
File Name: OCF

File Description: Orders carried forward

Contains the customer order load per period for each product.

File Content: OCF (a, b)

a. = product key
b. = 1 .... overdue load
    2-7 .. forward load for periods 1 to 6 inclusive

File Size: OCF (10, 7) = 70

Program Segments: QPLAN, QPLANPRINT, QTREVENTS

File Type: Integer

Special Notes: None
File Name: OCFWKS

File Description: Workfile

Contains the orders carried forward (order load) in equivalent weeks worth of production.

File Content: OCFWKS (a)

\[ a = 1-7 \] .. gross load in weeks for each period

File Size: OCFWKS (7) = 7

File Type: Integer

Program Segments: QPLANPRINT

Special Notes: Used to present information on the equivalent weeks worth of order load for each period in QPLANPRINT.
File Name: OFP

File Description: Off-line Plan

Contains off-line (assembly) plan for each product.

File Content: OFP (a, b)

- a = product key
- b = 1 .... not used
- .2-7 .. off line plan for periods 1 to 6 inclusive

File Size: OFP (10, 7) = 70

File Type: Integer

Program Segments: QPLAN, QPLANPRINT, QTREVENTS, WEEKLYPLAN.

Special Notes: None
File Name: OLDSCHED

File Description: Workfile

Temporary file used to retain the previous schedule when formulating a new supplier schedule.

File Content: OLDSCHED (a)

\[ a = \begin{cases} 1 & \text{schedule-arrears} \\ 2-13 & \text{monthly call-off} \end{cases} \]

File Size: OLDSCHED (13) = 13

File Type: Integer

Program Segments: SCHED

Special Notes: None
File Name: ONP

File Description: On-line plan

Contains the on-line (material) plan for each product.

File Content: ONP (a, b)

\[ a = \text{product key} \]
\[ b = 1 \ldots \text{not used} \]
\[ 2-7 \ldots \text{on-line plan for periods 1 to 6 inclusive.} \]

File Size: ONP (10, 7) = 70

File Type: Integer

Program Segments: STOCKMODEL, QPLAN, QPLANPRINT, QREVENTS, QTRRESET

Special Notes: None
File Name: OPFILE

File Description: Orders placed file

Contains detail of each customer order, including data relating to product requirement, due dates and state of completion.

File Content: OPFILE (a, b, c, d)

a = file address
b = item number (up to 5)
c = delivery phasing (up to 5 batches per item)
d = 1 .... product type
   2 .... quantity required (total)
   3 .... week due (actual)
   4 .... quantity unallocated
   5 .... quantity to be taken from stock
   6 .... quantity in an allocation
   7 .... quantity on-line
   8 .... quantity in test
   9 .... quantity in despatch
  10 .... order number

File Size: OPFILE (125, 5, 5, 10) = 31250

File Type: Integer

Program Segments: STOCKMODEL, ALLOCATE, DELPERF, DESPATCH, FESALL, ISSWIP, LOAD, MONTHEVENTS, OFFLINE, OPFCREATE, OPFPRINT, QTREVENTS, QTRRESET, STOCKVAL, WEEKEVENT.

Special Notes: OPFILE contains all live (i.e. not fully despatched) customer orders and is organised in a quasi-random sequence. Early orders will tend to be sequential, later additions will be inserted in the first available free position. Thus, the file will tend to have a high occupancy at the head, thus improving processing time.
The file is organised into two partitions; static data describing the order content (products, quantities, due dates) and dynamic data reflecting the state of completion of the order (status information).

Orders are removed from the file and the position is made available for new orders when the final delivery of all item lines has been made.
File Name: ORDBK

File Description: Order Book

Contains a summary of the orders due per period for each product.

File Content: ORDBK (a, b, c, d)

a = product key
b = week due (relative)
c = position in file
d = 1 .... order number
   2 .... item number
   3 .... quantity outstanding
   4 .... 0 = due
          1 = overdue
   5 .... 1 = part shipment permissible
          2 = no part shipment

File Size: ORDBK (5, 52, 15, 5) = 19500

File Type: Integer

Program Segments: STOCKMODEL, ALLOCATE, DELPERF, DESPATCH, FESALL, LOAD, LOADPLAN, MONTHEVENT, OPFCREATE, ORDBKPTR, QTREVENTS, QTRRESET, WEEKEVENT

Special Notes: The file is reset each quarter by advancing each data element by thirteen weeks. Orders with quantities outstanding in the first thirteen weeks prior to re-setting are overdue. To reduce file space, overdue orders are "tagged" and re-located at the tail of the file. Thus, an order which was originally located in period 8 and would normally be relocated to (8 - 13) = -5 is placed in period (39 + 8) = 47.

Orders are removed from the file when all item lines have been despatched.
File Name: ORDHIST

File Description: Order History

Contains monthly orders received history for use in forecasting.

File Content: ORDHIST (a, b)

\[ a = \text{product key} \]
\[ b = 1-12 \ldots \text{monthly orders received volume} \]

File Size: ORDHIST (5, 12) = 60

File Type: Integer

Program Segments: STOCKMODEL, FORECAST, LOAD, MONTHEVENT, OHRESET, QTREVENTS, WEEKEVENT.

Special Notes: File is reset each month by advancing each data elements by one period and clearing the final period. Orders received data is captured from sub-routine LOAD and augments the final period in ORDHIST.
File Name: ORDPERF

File Description: Order performance.

Contains statistics relating to the delivery performance achievement at complete order level.

File Content: ORDPERF (a, b)

\[ a = \begin{cases} 1 & \text{part shipment order} \\ 2 & \text{no part shipment order} \end{cases} \]

\[ b = 1-30 \] frequency histogram over the range -20 to +9

File Size: ORDPERF (2, 30) = 60

File Type: Integer

Program Segments: STOCKMODEL, DELPERF, DESPATCH, WEEKEVENT, WEEKPRINT

Special Notes: File contains two parts, part shipment order and no part shipment order statistics, each containing 30 intervals.

For part shipment orders, despatch of every item line on an order with a common due date will increment the appropriate interval by one.

For no part shipment, all item lines on an order must be completely despatched for the file interval to be incremented. Delays over the range -20 to +9 weeks are converted to the range 1 to 30 as data is captured and reconverted to represent the correct delay factor for subsequent analysis.
File Name: ORF

File Description: Orders received forecast

Contains orders received forecast consolidated into quarterly buckets.

File Content: ORF (a, b)

\[ a = \text{product key} \]
\[ b = 1 \ldots \text{not used} \]
\[ .2-7 \ldots \text{forecast for periods 1 to 6} \]

File Size: ORF (10, 7) = 70

File Type: Integer

Program Segments: FORECAST, QPLAN, QPLANPRINT, QTREVENTS

Special Notes: None
File Name: PA

File Description: Production to be allocated.

Contains the planned allocation of equipments from assembly or stock into test.

File Content: PA (a, b)

\[ a = \text{product key} \]
\[ b = 1 \ldots \text{not used} \]
\[ 2-7 \ldots \text{production to be allocated for periods 1 to 6 inclusive} \]
\[ 8-10 \ldots \text{contains duplicate of period 6} \]

File Size: PA (10, 10) = 100

File Type: Integer

Program Segments: QPLAN, QPLANPRINT, QTREVENTS, WEEKLYPLAN

Special Notes: For planning purposes, the test load is assumed constant, therefore the production to be allocated may be considered as the equivalent of the delivery plan.

The content of periods 8 through 10 is used for data presentation purposes in QPLANPRINT.
File Name: PIPELINE

File Description: Orders Pipeline

Contains orders received from the order generator pending order promising (loading).

File Content: PIPELINE (a, b, c)

\[
\begin{align*}
    a & = \text{position in file} \\
    b & = 1 \ldots \text{order header} \\
    c & = 1 \ldots \text{order number} \\
        & \quad 2 \ldots 1 = \text{part shipment permissible} \\
        & \quad 2 = \text{no part shipment} \\
        & \quad 3 \ldots \text{number of items} \\
    2-6 & \ldots \text{Item detail} \\
    c & = 1 \ldots \text{item number} \\
        & \quad 2 \ldots \text{product type} \\
        & \quad 3 \ldots \text{quantity} \\
        & \quad 4 \ldots \text{value}
\end{align*}
\]

File Size: PIPELINE (20, 6, 4) = 480

File Type: Integer

Program Segments: STOCKMODEL, INITODS, LOAD, ORDERS, WEEKEVENT

Special Notes: Temporary file used to organise orders introduced by the order generator in preparation for the order promising (order loading) process. When orders have been successfully loaded, order detail is transferred to OPFILE and summarised in ORDBK.
File Name: PRODMIX

File Description: Product Mix

Contains product mix data modified for trend prior to updating the product mix table (Table 6).

File Content: PRODMIX (a)

a = product key/data

FileSize: PRODMIX (7) = 7

File Type: Real

Program Segments: STOCKMODEL, QTREVENTS, TRENDI

Special Notes: The file is initially loaded from Table 6 with the starting mix values. Subsequent application of trend will modify PRODMIX values, which must be corrected to the range 0 - 100 before updating Table 6.
File Name: R

File Description: Recommended production rates

Contains a sequence of fixed production rates for each product.

File Content: R (a, b)

a = product key
b = 1-10 .... Series of rates (units/week) in ascending order

File Size: R (10, 10) = 100

File Type: Integer

Program Segments: STOCKMODEL, QPLAN, QPLANPRINT, QTREVENTS

Special Notes: None
File Name: REQFILE

File Description: Requirements File

Contains the weekly gross requirements for each item.

File Content: REQFILE (a, b)

a = product key
b = 1-52 ... weekly requirements

File Size: REQFILE (25, 52) = 1300

File Type: Integer

Program Segments: STOCKMODEL, ALLOCATE, EBQ, EXPLODE, OFFLINE, QPLAN, QTREVENTS, SCHED, SUBPROG, WEEEVENT, WEEKLYPLAN

Special Notes: File is recreated each quarter as a result of the quarterly planning and requirements calculations routines.
File Name: RR

File Description: Running Rate

Contains the selected production rate per period for each product.

File Content: RR (a, b)

a = product key
b = 1 .... not used
2-7 .. production rate for periods 1 to 6 inclusive
8-11 . not used

File Size: RR (10, 11) = 110

File Type: Integer

Program Segments: QPLAN, QPLANPRINT, QTREVENTS

Special Notes: None
File Name: SCHEDULE

File Description: Supplier Schedule

Contains monthly delivery requirements from the supplier, including a gross arrears value.

File Content: SCHEDULE (a, b)

\[ a = \text{item key} \\
\[ b = 1 \ldots \text{arrears value} \\
\[ 2-13.. \text{monthly requirements for periods 1-12} \]

File Size: SCHEDULE (25, 13) = 325

File Type: Integer

Program Segments: STOCKMODEL, ARRIVAL, QTREVENTS, RECEIVE, SCHED, WEEKEVENT

Special Notes: SCHEDULE contains the gross monthly delivery requirements from the supplier and includes a simple gross value for the outstanding arrears. Actual deliveries against the schedule are derived from subroutines RECEIVE and located in file INPUT.
File Name:             STKORD

File Description:  Stock Order Number

Contains the last stock order number allocated for each item type.

File Content:        STKORD (a)

a = item key/last order number assigned

File Size:           STKORD (20) = 20

File Type:           Integer

Program Segments:    ALLOCATE, WEEKEVENT

Special Notes:       None
File Name: SUBFILE

File Description: Sub-assembly Programme

Contains the sub-assembly programme derived from the requirements file and incorporating economic batch rules.

File Content: SUBFILE (a, b)

\[ a = \text{item type} \]
\[ b = 1-26.... \text{weekly sub-assembly programme} \]

File Size: SUBFILE (12, 26) = 312

File Type: Integer

Program Segments: STOCKMODEL, ALLOCATE, MONTHEVENT, SUBPROG, WEEKEVENT

Special Notes: SUBFILE contains the schedule of sub-assembly deliveries to stock and is the basis for material allocation and subsequent manufacture. The schedule is derived from REQFILE and is modified to take account of economic batch rules, buffer stock and shortages.
File Name: SUBQ

File Description: Sub-assembly Queue

Contains a queue of sub-assembly orders in manufacture.

File Content: SUBQ (a, b)

  a = position in file
  b = 1 .... part number
      2 .... due date (actual)
      3 .... quantity
      4 .... timer

File Size: SUBQ (50, 4) = 200

File Type: Integer

Program Segments: STOCKMODEL, ISSWIP, OFFLINE, OPFCREATE, SQPRINT, WEEKEVENT

Special Notes: File SUBQ represents the queue of work in the sub-assembly manufacturing department. The queue contains a mixture of sub-assembly types and may be organised in "first-in-first-out" or due date priority by utilising a sort pre-processor. The file is consolidated after all order completions have been processed in subroutine OFFLINE.
File Name: TABLE 1
TABLE 2
TABLE 3
TABLE 5
TABLE 6
TABLE 7
TABLE 8
TABLE 11

File Description: Sampling Tables
Contains histogram definition for use in sampling routines.

File Content: Table (a, b)

a = position in file
b = 1 .... parameters
2 .... cumulative frequency

File Size: TABLE (7, 2) = 14 (for TABLE 1 through TABLE 7)
TABLE (2, 2) = 4 (for TABLE 8, TABLE 11)

File Type: Integer

Program Segments:
TABLE 1 - STOCKMODEL, QTREVENTS, RECEIVE
TABLE 2 - STOCKMODEL, QTREVENTS, RECEIVE
TABLE 3 - STOCKMODEL, EXPEDITE, WEEKEVENT
TABLE 5 - STOCKMODEL, INITODS, ORDERS, WEEKEVENT
TABLE 6 - STOCKMODEL, INITODS, ORDERS, QTREVENTS, TRENDI, WEEKEVENTS
TABLE 7 - STOCKMODEL, INITODS, ORDERS, WEEKEVENT
TABLE 8 - STOCKMODEL, INITODS, ORDERS, WEEKEVENT
TABLE 11 - STOCKMODEL, ARRIVAL, WEEKEVENT

Special Notes: Files (TABLES) contain static data which is pre-defined for each simulation run. The histograms held in each table are cumulative over the range 0-100. The general format is as below:

Parameter  1  2  3  4  5  6  7
Cum. freq. 10 20 35 50 75 90 100
This equates to a frequency histogram as depicted below:

![Frequency Histogram](image)

File content is as follows:

TABLE 1 = Normal input - profile of material deliveries around the scheduled receipt date.
TABLE 2 = Arrears input - profile of schedule arrears receipts.
TABLE 3 = Expedited input - profile of the relative success of expediting action.
TABLE 5 = Number of items on the customer order.
TABLE 6 = Product type of selected item on order.
TABLE 7 = Quantity of product per line item on order.
TABLE 8 = Part ship/no part ship indicator.
TABLE 11 = Reject input - profile of reschedule for material rejects.
File Name: TESTQ

File Description: Test Queue

Contains a queue of products which have been moved into the test department pending despatch.

File Content: TESTQ (a, b, c)

a = product key
b = position in file
c = 1 .... order number
   2 .... item number
   3 .... due date
   4 .... quantity
   5 .... timer

File Size: TESTQ (5, 50, 5) = 1250

File Type: Integer

Program Segments: STOCKMODEL, DESPATCH, FESALL, OFFLINE, OFFCREATE, TQPRINT, WEEKEVENT

Special Notes: File is organised into a separate queue per product, each representing a manufacturing resource area.

New entries will be added to the tail of the file and the file is accessed sequentially from the head. Thus, the implicit priority sequence is "first-in-first-out". The file may be sorted by due date (utilising a sort pre-processor) if a due date priority is required.
File Name: TOTAL

File Description: Total Load

Contains the gross order load per week for each product.

File Content: TOTAL (a, b)

a = product key
b = 1-52 .... order load per period

File Size: TOTAL (5, 52) = 260

File Type: Integer

Program Segments: STOCKMODEL, DESPATCH, LOAD, OPFCREATE,
QPREVENTS, QTRRESET, WEEKEVENT

Special Notes: File is augmented as orders are loaded in subroutine LOAD and decreased as orders are moved into despatch in subroutine DESPATCH.
File Name: TRENDS

File Description: Trend Data

Contains the order mix trend for each product.

File Content: TRENDS (a)

a = product key/data

File Size: TRENDS (7) = 7

File Type: Real

Program Segments: STOCKMODEL, QTREVENTS, TRENDS

Special Notes: The file contains a parameter for each product, used to modify the product mix ratios in the order generator sampling routines. Thus the relative proportion of the total orders received for each product may be changed without affecting the overall gross volume. The trend data is the per unit increase (or decrease) per quarter.
File Name: WIP

File Description: Work in Progress

Contains planned level of assembly work in progress per quarterly period.

File Content: WIP (a, b)

    a = product key
    b = 1 .... opening w.i.p. (actual)
        2-7 .. planned closing w.i.p. for periods 1 to 6.

File Size: WIP (10, 7) = 70

File Type: Integer

Program Segments: QPLAN, QPLANPRINT, QTREVENTS

Special Notes: None
File Name: WORKFILE

File Description: Workfile

Contains transient information used in the plan explosion netting process.

File Content: WORKFILE (a)

\[ a = \text{value per period} \]

File Size: WORKFILE (52) = 52

File Type: Integer

Program Segments: EXPLODE

Special Notes: None.
DATA DICTIONARY

The data dictionary describes the variables contained within the program segments as an aid to maintenance and program modification.
DATA DICTIONARY

ACTWK  Actual week (simulated)
ALLQTY  Quantity in allocation status
AVAIL  Calculated available stock
AVGE  Average
BUFFER  Calculated buffer level
BUFFERWKS  Weeks worth of buffer
CAP/CAPACITY  Capacity
COMPVAL  Component value
COMPBUFR  Component buffer level (planned) in weeks
CSVAL  Commercial stock value
CUMDIF  Cumulative difference between new and previous material plans
CVARN  Co-efficient of variance
DATO  Actual calendar date
DATEREQD  Date required
DDATE  Due date
DELAY  Delay between current week and due date
DEMAND  Calculated period demand
DUE (DATE)  Due date (general application)
EQUIBUFR  Equipment buffer level (planned) in weeks
FESVAL  Finished equipment stock value
FLAG  Indicator used in quarterly planning process
ICOUNT  Number of items required
IPRINT  Print key set by diagnostic file
ISEEDI-12  Random number seeds
ISHORT  Sort key, used to determine priority rules
ITEMCOUNT  Number of order item lines
ITEMNO  Order line item number
MODE  Planning mode (order, stock, mixed)
OCFMIN  Minimum value of orders carried forward, expressed in weeks
OLDBUFFER  Previous buffer level
ORDCOUNT  Counter indicating the next available customer order reference
ORDNO  Customer order number
PASUM: Sum of production to be allocated
PCBBUFR: Printed circuit board buffer level (planned) in weeks
PROD: Product key
QTY: Quantity (general application)
REFWEEK: Reference week, used to derive the simulated calendar week (or actual week)
REJECT: Quantity of rejected batch
REQD: Quantity required
RESPONSE: Schedule response time in months
RUNTIME: Runtime for each experiment in simulated weeks
SHORT: Calculated component shortage
STOCK: Opening stock
SUBVAL: Sub-assembly value
TESTWIP: Test work in progress value
TIME: Not used
TOTVAL: Total value
TYPE: Product key
VALUE: Value selected from histogram
WEEK: Simulation week, relative to first period - reset each quarter
WIP: Work in progress
WIPVAL: Work in progress value (assembly)
WKCAP: Weekly capacity (derived from quarterly value)
XSTOCK: Excess stock