MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFIT/CI/NR- -

AD-A170 671

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFIT/CI/NR-86- 91T	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle) Enhanced Simulation Output Through The Use Of Spreadsheet Software	5. TYPE OF REPORT & PERIOD COVERED THESIS/DISSERTATION
	6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s) John Patrick Dieltz	8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: Iowa State University	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

1. CONTROLLING OFFICE NAME AND ADDRESS AFIT/NR WPAFB OH 45433-6583	12. REPORT DATE 1986
	13. NUMBER OF PAGES 86

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLAS
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

"Original contains color plates: All DTIC reproductions will be in black and white"

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-1

LYNN E. WOLAVER 6 Aug R
Dean for Research and
Professional Development
AFIT/NR

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

DTIC
ELECTE
AUG 1 2 1986

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED.

DD FORM 1473 1 JAN 73   EDITION OF 1 NOV 65 IS OBSOLETE

86   8   12   027

Enhanced simulation output through

the use of spreadsheet software

by

Jonn Patrick Dieltz

A Thesis Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major: Industrial Engineering

Approved:

Thomas A. Barta
In Charge of Major Work

Harold L. McPhul
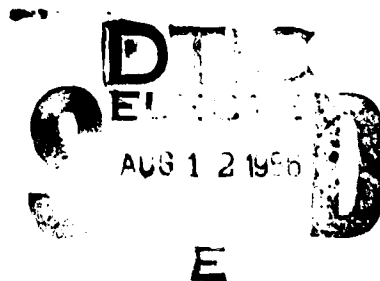For the Major Department

O. J. Morano
For the Graduate College

Iowa State University
Ames, Iowa

1986

ABSTRACT

This paper describes the development of a technique to
display microcomputer simulation output data graphically,
using spreadsheet software. The simulation data was
generated by GPSS/PC[1] simulation software and displayed
using Lotus 1-2-3[2] spreadsheet software. Snapshots of data
were taken at regular intervals during the simulation. The
simulation output data files were then translated into Lotus
1-2-3 worksheet files by use of the Data Interchange Format
(DIF). Once translated into the worksheet format the
snapshots of data were combined onto a single worksheet
which contained macros to generate the graphs.

Macros were written to generate and save each graph
setting. A separate macro was written for queue statistics,
utilization statistics, and a schematic presentation of the
problem. For later display, other macros were used which
recalled the previously stored graph settings. The snapshot
method can, if the time interval between data summaries is

---

[1] Lotus 1-2-3 is a registered trademark of Lotus
Development Corporation.

[2] GPSS/PC is a registered trademark of Minuteman
Software.

small enough, reach a nearly animated condition when the graphs are recalled.

All types of Lotus 1-2-3 graphs were used to display the simulation output; bar charts, line graphs, XY plots, and pie charts. Each macro generated a series of graphs which showed how the the queue or utilization statistics of the model changed during each snapshot interval. The XY plot was also used to generate a schematic diagram which included combined queue and utilization statistics of the system being modeled.

Lotus 1-2-3 offers the modeler many graph types to present simulation results in a format best suited to the particular data. The graphics can be attractively presented using output from proven discrete event simulation languages, such as GPSS and SLAM, without buying expensive graphics software tailored to a specific language.

TABLE OF CONTENTS

INTRODUCTION

Use of microcomputers is the latest trend in simulation. Mainframe languages have either been adapted for use on microcomputers or are in the process of being rewritten for micros. Increases in the use of microcomputers in the workplace and the availability of larger memory capacity on the latest micros make microcomputer simulation a possibility for organizations of all sizes. Some of the language adaptations are transparent to those experienced in the language, while in some cases, the micro versions of simulation languages make use of the interactive capabilities of micros.

Computer simulation output data contain detailed information that may not be apparent to anyone other than the computer professional or engineer who developed the simulation model. This presents a problem for the modeler when presenting the results of a simulation analysis to the decision makers in management. Key points from the simulation model may be lost in the printed output simply because the data was not in an easy to understand format.

Graphics packages are available for some mainframe simulation languages which allow the modeler to select

different kinds of displays for the various types of data in the model. Mainframe languages may also offer animated graphics or the ability to interface with other packages or languages which can animate the output data. These various methods offer the simulation modeler the flexibility to present the model output data in a form that is easily understood by those not directly involved with the simulation modeling.

A look through current microcomputer magazine or journal advertisements indicates a trend toward graphic output displays for almost any kind of data. These packages cover a broad range in both sophistication and cost. The criteria for this project were to get the simulation data in an interesting and readable format and do it at minimum cost.

The minimum cost criterion required the use of a readily available package with graphics capability. Lotus 1-2-3 satisfied this criterion. It is an integrated spreadsheet and data management software package with graphics that has enjoyed widespread popularity in business and industry since its introduction in 1983 (14).

At the time this project was beginning only one of the

available microcomputer simulation languages had a graphics
capability (3). This language required the use of a
specific model microcomputer which was not available for
this investigation. The simulation language used, GPSS/PC,
will run on an IBM or compatible microcomputer. Attempts to
use it on another microcomputer were unsuccessful because a
unique plug-in security device is required to run the
simulation.

REVIEW OF PERTINENT LITERATURE

What is the advantage of displaying simulation data in graphical form? The simple answer is communication. In other words, graphs are an easy to understand way of presenting some of the complex relationships that exist in a simulation problem (19). They may also be used to call attention to a particular aspect of a simulation while the normal printed output may be used to study the details (16). Graphics put the old phrase "a picture is worth a thousand words" to work.

Simulation, to some, is like a foreign language (6). The output of the simulation analysis may be equally as foreign. Using graphics with the simulation facilitates understanding of the data by decision makers. Taking the mystery out of the simulation process helps both the modeler and the decision makers in understanding each other and using simulation of plant operations to best advantage. Ford Motor Company has found this to be the case in their European and North American operations (6, 16).

The precedent for using graphic display of simulation data was established with mainframe computers. Numerous graphics packages are available for use with mainframe and

minicomputer simulation languages (19, 27, 29). Some of
these such as TESS (27) and IDEF2 (29) work only with a
specific simulation language. Other packages, such as AID
(19) and SIMCHART (19), can be used with any simulation
model.

These graphics packages all give the modeler the
capability to display simulation data in histograms, bar
charts, pie charts, and plots. TESS and IDEF2 also allow
the modeler to construct the simulation model using graphics
(27, 29). Kachitvichyanukul and Lloyd (10) state that TESS
is the only truly integrated simulation software system
available for mainframe applications. An integrated
simulation system is a software system which supports the
pre-simulation model development with statistics collection,
the actual model entry with graphics or other support, and
the post-simulation data handling with report generators and
graphics. They argue that the lack of integrated systems
for model development and data display takes a large amount
of the modeler's time and skill away from the actual
simulation.

Real-time display of the simulation is also possible
with the use of animated graphics. Animation is also cited
as one way of getting decision makers to see that model

characteristics are valid (6).

The advent of tne microcomputer brought investigations
into its use for simulation. Early attempts were confined
to the 8-bit processors with limited amounts of Random
Access Memory (RAM) (16). Later, 16-bit processors with
increased RAM made more complex simulations possible. Ford
of Europe uses a two-stage modeling process to alleviate
size and speed problems associated with micro based
simulation (15). They generate what they call "data
packets" from models of sections of the plant to be fed into
models of other sections down stream in the production
process. The complexity of this system requires complicated
models to be done by their Operations Research Group. Their
industrial engineering staff is limited to developing simple
models.

Kacnitvicnyanukul and Lloyd (10) took another approach
to using microcomputers in simulation. They used a
relational data base management system as a central point in
what they call an integrated simulation support system.
They used the data base management system to collect the
pre-modeling data and to store and display the post
simulation data.

Another study by Vasudev and Pruett (28) focused on a comparison between micro and mainframe simulation capabilities. Using SIMBASIC and SIMPAS, simulation languages based on BASIC and PASCAL, they compared the results of 8-bit and 16-bit microcomputers with those of a 32-bit mainframe. The study noted that while the actual data from each computer system varied, they fell into a range which was reasonable enough for sound decision making. The accuracy of another microcomputer simulation done by Mourant (22) also compared favorably with results obtained from an established simulation language on a mainframe. In other words, used individually, the data would have led the decision maker to reach the same conclusion.

Having established the accuracy of the data from the different computer systems, Vasudev and Pruett (28) went on to look at how the systems compared in terms of timeliness and cost-effectiveness. When timeliness was measured in terms of Central Processor Unit (CPU) time, the mainframe computer was the clear winner. However, when other factors such as accessibility and time sharing were considered the microcomputers compared favorably. The study concluded that if mainframe computer costs are in the $500 to $1000 per CPU hour range, microcomputer based simulation is more cost-effective. Favreau, Murtha, and Marr (5) state that

the speed of execution for microcomputers is irrelevant. Their reasons are that a micro is a single-user system which provides quicker feedback with an execution cost of almost zero.

While the criteria in the Vasudev and Pruett (28) study; accuracy, timeliness, and cost; favored the microcomputer, one shortcoming was noted. At the time of the study, the more commonly used simulation languages were not available for microcomputers. The authors noted that as simulation languages are adapted for microcomputers and as the capabilities of the microcomputer increase, microcomputer simulation will be more and more common (28).

Since the above mentioned study, some of the most widely used mainframe languages; GPSS, SIMSCRIPT, and SLAM (1); have become available for microcomputers. One of these, GPSS/PC, is the microcomputer version of General Purpose Simulation System (GPSS). In its conversion to the micro, a built-in program editor and a system of commands were added which allow interactive modeling techniques (12). Due to limitations of microcomputers, certain features of the mainframe language were omitted on GPSS/PC. These features deal primarily with interfacing the simulation program with other computer languages for subroutine

manipulation of data or the output of data to separate files
(12).

Tnese shortcomings are somewnat offset by the addition
of features which capitalize on the interactive environment
of the microcomputer. The most notable feature is the
addition of tne program eaitor. Tne on-screen eaitor checks
each model statemenc as it is entered for syntax errors
(12). The interactive structure of the program also allows
the modeler to change model cnaracteristics by adding or
deleting blocks or changing values during a modeling
session. The effects of changes can be seen by using a PLOT
statement to view a System Numerical Attribute (SNA) during
a simulation run (8, 12).

Favreau, Murtha, and Marr (5) have stated that
simulation software systems have lagged behind tne
development of other areas in microcomputers. They suggest
that a spreadsheet be one of the packages available to the
microcomputer simulation modeler. Indeed, spreadsheet
programs have been used for many types of problems including
simulation analysis. The keyboard macro commands of Lotus
1-2-3 provide tne user with an ability to write programs
(20, 25). The macros are as powerful as a computer language
and have been called tne most powerful programming language

in business today for small applications (7). The key advantage of using spreadsheet macros is that they don't require an extensive programming background (17).

Spreadsheet simulations have been done using macros, but the macros lack some of the built-in functions of a simulation language (25). A simple step performed in a simulation language such as establishing a queue requires a number of operations on a spreadsheet. This is not to say that the final results obtained are any less accurate. Masri and Moodie (17) compared the results from a flow shop simulation done on a spreadsheet with those from a simulation language model and found them to be similar.

Even though Lotus 1-2-3 has some shortcomings in the actual simulations, its graphics capability has won high praise (17, 25). Its ability to convert statistics into graphs makes graphing a "thinking tool" (20). Reese (25) has suggested that future simulation languages adopt the graphic capability of a program like Lotus 1-2-3.

Spreadsheets are used to solve a broad range of problems. They are being used in such diverse applications as circuit design (9), network analysis (2), to project space requirements (23), scheduling (20), manpower planning

(20) and a host of various types of financial analysis (20).
Spreadsheets can be used on such diverse types of problems
because the solutions rest on the data relationships. These
relationships can be described by defining the relationships
between the cells in the spreadsheet. The process of
defining the cellular relationships has been called
declarative programming (11). This is much different from
the procedural steps of programming languages such as
FORTRAN or BASIC. The declarative programming done on the
spreadsheet is very straight forward and doesn't require the
user to learn all of the intricacies of a procedural
language. The relative ease of using the spreadsheet on
such a vast range of applications has led to spreadsheets
being the most widely used computer applications program
(11).

Lotus 1-2-3 is one of the most popular spreadsheet
programs (14). One reason is that it has more capabilities
than other spreadsheets (20). It offers a unique
combination of a spreadsheet, built-in special functions,
data management, and graphics. As stated above, the
keyboard macros and special functions also give it an
extensive programming capability. The macros and special
functions have been expanded in the newest version just
released (13). In a survey of 1-2-3 users, Lotus has found

that the graphics are one of the least used portions of the package (18).

Lotus is unique in another aspect. It is probably the only software package that is now supports its users with their own magazine (2, 11, 13). The magazine offers articles on ways to best use the capabilities of 1-2-3 as well as giving readers advice on ways to overcome problems they have encountered using the software. Lotus 1-2-3 has also spawned an entire industry of support products. Issues of Lotus and Byte magazine contain many advertisements for products made to work with Lotus 1-2-3.

Graphic presentation of simulation results aids management's understanding of the simulation process. Simulation on microcomputers is a way of putting simulation within the reach of almost any organization. Lotus 1-2-3 is a widely available software package which offers a way of displaying the output data of microcomputer simulation in graph form.

OBJECTIVE

Tne purpose of this study was to investigate the capabilities of a spreadsheet program to display the output of simulation models on a microcomputer. Lotus 1-2-3[1] was the spreadsheet used to display GPSS/PC[2] simulation output data. Real-time animated graphics will not be possible due to GPSS/PC's inability to interface witn other language subroutines or data files during simulation. A series of stop-action "snapshots" of the data can be taken and then presented in sequence on 1-2-3 to give a semi-animated display.

The investigation was conducted with Lotus 1-2-3 version 1A and GPSS/PC version 1.1.

---

[1]Lotus 1-2-3 is a registered trademark of Lotus Development Corporation.

[2]GPSS/PC is a registered trademark of Minuteman Software.

PROCEDURES

The procedural steps discussed in this section can be divided into two areas. First, a discussion of the GPSS/PC steps used to generate the data files from the model and, second, the Lotus 1-2-3 procedures used to produce the graphics.

GPSS/PC

The steps used will be illustrated by an example problem. Before the actual development, however, the unique aspects of GPSS/PC language will be discussed. This discussion is necessary because some of the differences in this version of GPSS were limiting factors in the final outcome.

GPSS/PC lacks the blocks normally found in other versions of the language which allow interface with outside storage media. Specifically, the CHANGE, JOBTAPE, REWIND, READ, WRITE, and PRINT blocks are not found in GPSS/PC. These blocks allow the modeler to work with tape or other storage devices to write data to outside files. These files could then be accessed by specially designed subroutines to generate animated graphics or unique reports.

Another missing block is the HELP block. This block is
used to send data directly to subroutines for specially
required manipulation. This is useful in cases when there
isn't a GPSS block which accurately describes the process
being modeled. The ability to access outside subroutines
increases the modeler's flexibility in applying the language
to a given problem. The HELP block is another way the
modeler could pass control to an outside subroutine to
generate the data necessary to produce animated graphics.
Lack of a HELP block also prevents any real-time processing
of the data and limits the modeler to post-simulation report
data.

The result of the omission of the above blocks from the
microcomputer version of GPSS limits the modeler to using
the data generated in the standard output summary. There
aren't any provisions in the language for generating
specialized report formats. The modeler is limited to the
standard report supplemented by tables, savevalues, matrix
savevalues, and functions. The modeler can use these to
record any of the System Numerical Attributes (SNAs)
desired.

Another feature omitted in this version of the language

is the Snap Interval Counter. The "C" operand of the START statement is inoperative. This prevents the modeler from automatically getting intermediate data summaries at selected intervals. This can be overcome as illustrated in the example below by using a series of the REPORT commands successively with START commands to obtain the desired interval of reporting (8).

While there are the omissions from the language as noted above, there are also some additions worthy of mentioning. The foremost of these is the on-screen editor. This facilitates model development by checking each block statement for syntax as it is typed. As each entry is made the cursor advances to the next required entry in the correct column. If an error is made in the entry the cursor will not advance until the correction is made. It also requires that only the first three letters of a block statement need be entered. For example, if the modeler wishes to place a GENERATE block in the model, ne only needs to type GEN followed by a space. The editor completes the word and advances to the correct column for the A operand.

The EDIT, RENUMBER, DISPLAY, and DELETE commands are used in model development to edit the model statements (8). Model statements can be entered in two different ways. The

statements can be completely interactive, in which case they are entered without line numbers. Each line is then executed as it is entered. The model can also be developed as a "savable program" by numbering each line. The SAVE command saves the numbered model statements to an ASCII format disk file. The model can also be developed with a savable "core" along with some unnumbered control statements used to start and stop the simulation or vary certain parameters. After the model is developed, statements can be changed with the EDIT or DELETE command, or by substituting a new statement with the same line number as the previous statement.

Other statements unique to GPSS/PC are the STOP and CONTINUE commands (8). The STOP allows the modeler to insert a condition in the model which will stop the simulation. This is one of the features which capitalize on the interactiveness of the microcomputer. When the stop condition is reached, the modeler may modify statements within the model and use the CONTINUE command to resume the simulation.

The PLOT (8) command gives the modeler limited ability to view the simulation in real-time. A selected SNA can be viewed on an XY graph during a simulation run. Time is

displayed on the x-axis and the value of the desired SNA is
plotted on the y-axis. The SNA values are plotted over time
as the simulation develops. Up to two PLOT statements can
be viewed at a time on a split screen. The data are
displayed on the screen only and not part of the standard
data report.

Online help is also available during model development.
Simply pressing the [?] key will call up a help screen which
will display the options for the entry being considered (8).

The simulation problem used to develop the graphics
procedures is found in Schriber (26) example 2F:

"Assembled television sets move through a
series of testing stations in the final stage of
their production. At the last of these stations,
the vertical control setting on the sets is
tested. If the setting is found to be functioning
improperly, the offending set is routed to an
adjustment station, where the setting is modified.
After adjustment, the television set is sent back
to the last inspection station, where the setting
is again inspected. Television sets passing the
final inspection phase, whether the first time or
after one or more routings through the adjustment

station, pass on to a packing area."

"Television sets arrive at the final
inspection station from the previous station every
5.5 plus or minus 2 minutes. Two inspectors work
side-by-side at the final inspection station. The
time required to inspect a set is 9 plus or minus
3 minutes. About 85 percent of the sets pass
inspection and continue on to the packing
department. The other 15 percent are routed to
the adjustment station, which is manned by a
single worker. Adjustment of the vertical control
setting requires 30 plus or minus 10 minutes."

The simulation was performed to determine the amount of
staging space required for television sets waiting for
inspection and adjustment. Pritsker (24) also uses this
problem as Example 6-2 and asks the modeler to determine how
long the television sets are in this final inspection and
adjustment station. The simulation code to determine the
required staging space and throughput time is in Figure 1.

The savable program code of Figure 1 does not include
control statements because control is maintained
interactively by issuing commands from the keyboard during

20

the simulation session. The control statements in Figure 2 were entered unnumbered and executed immediately by tne program. The model was run for one week of simulated plant operation, as in Schriber (26).

```
10  ***********************************************************
20  *                                                         *
30  *           TV INSPECTION AND REPAIR                      *
40  *                                                         *
50  ***********************************************************
60  *
61  * Table Definition
62  *
63  RTABLE    TABLE        M1,150,150,5    ;Residence Time
64  *
70  * Storage Capacity
80  *
90  INSP      STORAGE      2               ;Two Inspectors
100 *
110 * Model Segment 1
120 *
130           GENERATE     55,20           ;T.V. Arrivals
140 STA1      QUEUE        LINE            ;Wait for Inspection
150           ENTER        INSP            ;Capture an Inspector
160           DEPART       LINE            ;Leave Waiting Line
170           ADVANCE      90,30           ;Inspection Time
180           LEAVE        INSP            ;Release Inspector
190           TRANSFER     .15,,STA2       ;15 % Failure Rate
195           TABULATE     RTABLE          ;Record Residence Time
200           TERMINATE
210 STA2      QUEUE        LIN2            ;Wait for Adjustment
220           SEIZE        ADJT            ;Capture Repairman
230           DEPART       LIN2            ;Leave Waiting Line
240           ADVANCE      300,100         ;Adjustment Time
250           RELEASE      ADJT            ;Free Repairman
260           TRANSFER     ,STA1           ;Return to Inspection
270 *
280 * Model Segment 2 - Time Unit = 0.1 minute
290 *
300           GENERATE     4800            ;8-hour day
310           TERMINATE    1
```

Figure 1. GPSS/PC model code

The RMULT statement was used to reseed the first random number generator prior to each run. This was done to insure the data from each run was generated by the same random number stream. The CLEAR statement was used to return the starting conditions to the same unused state prior to the run.

```
RMULT        111
REPORT       B:DAY1.GPS        ;DAY 1 SUMMARY
START        1
RMULT        111
CLEAR
REPORT       B:DAY2.GPS        ;DAY 2 SUMMARY
START        2
RMULT        111
CLEAR
REPORT       B:DAY3.GPS        ;DAY 3 SUMMARY
START        3
RMULT        111
CLEAR
REPORT       B:DAY4.GPS        ;DAY 4 SUMMARY
START        4
RMULT        111
CLEAR
REPORT       B:DAY5.GPS        ;DAY 5 SUMMARY
START        5
```

Figure 2. Supplementary control statements

The REPORT command assigned the unformatted simulation data to the file name declared in the command. The comment portion of the REPORT command, the text after the semicolon, contains the title GPSS/PC assigns to the report output.

The simulation session was ended by the END statement.

The result of the session was five unformatted data files which were then processed by the GPSS/PC report processing program. GPSSREPT processes the unformatted output data files into the standard report format. An example of the standard report format appears in Appendix A. The formatted reports can be displayed on the screen, sent to a printer, or saved in an ASCII format disk file from GPSSREPT.

After saving the formatted reports on disk files, they were translated to Data Interchange Format (.DIF) files using a GPSS/PC program called Differ. The .DIF file is a structured format used to exchange data files from one program to another (4). The .DIF is a standard way many programs use to exchange data from one to the other. It is used by spreadsheets such as Lotus 1-2-3 and VisiCalc and by other simulation languages, such as SLAM. Each .DIF file must have the four required header entries and end of file entry seen in Figure 3. An entire GPSS/PC formatted report converted to a .DIF file appears as Appendix C. Numeric values appear in the file as numbers while character strings are enclosed by quotation marks.

The original version of GPSS/PC used for this investigation had an error in the .DIF translation program. The TUPLES statement was missing and 1-2-3 could not

translate the file to the Lotus 1-2-3 worksheet format. A
TUPLE is a data record and the TUPLES statement tells the
program which translates the .DIF file to another format how
many records to translate. A .DIF file cannot be translated
without the TUPLES statement. It was possible to overcome
this by manually counting the number of records in the file
and putting a TUPLES in the correct place in the file with a
text editor.

```
.DIF statement        Explanation of Terms

TABLE                 -identifies file as .DIF
  0 , 1
 " "
VECTORS               -specifies number of vectors in file
  0 , 50                (entries between BOT)
 " "
TUPLES                -the length of each vector
  0 , 10                (number of tuples)
 " "
DATA                  -last header item, data follows
  0 , 0
 " "
-1 , 0
BOT                   -beginning of tuple
  .
  .
  .


EOD                   -end of data
```

Figure 3.   Example of .DIF header and end statements

Minuteman Software, the producer of GPSS/PC, was
notified of the error and made a correction to the program.
The revised version of the Differ program translates the
data files into .DIF files which contain the TUPLES

statement.

```
            _____
           |                   |
           |   GPSS  Model     |
           |_____|
                     |
            _____|_____
           |                   |
           |   Unformatted     |
           |   Data            |
           |   File            |
           |_____|
                     |
            _____|_____
           |                   |
           |   Formatted       |
           |   Data            |
           |   File            |
           |_____|
                     |
            _____|_____
           |                   |
           |   .DIF            |
           |   File            |
           |_____|
```
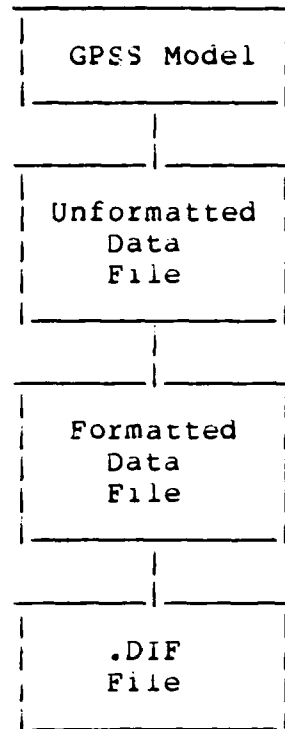
Figure 4. File processing summary

The process involved in getting from tne GPSS
simulation code to the .DIF file is summarized in Figure 4.
These steps do not have to be done sequentially for each
run.   In this case, the five output files were produced in
one simulation session.   After tne simulation session, the
five unformatted files were formatted and saved to disk
files.   An example of a formatted report file appears as
Appendix A and an unformatted report file appears as

Appendix B. The Differ program was then used to translate
the five formatted files to five .DIF files. During this
process none of the files from the previous step are
destroyed unless the modeler takes steps to delete them with
an operating system command.

LOTUS 1-2-3

Lotus 1-2-3 has five types of graphs for displaying
data in different ways. The pie chart can be used to
display only a single type of data while the bar, stacked
bar chart, and line graph can display up to six data ranges.
The XY graph can be used to display a variable which varies
with the values of a second variable, this is called two
dimensional data (15). Each graph type can be selected by
the modeler to display the desired data to best advantage.

The pie chart is the most limited since it can only
display a single set of data. It can, however, be a very
effective way of displaying that single set of data. Each
data entry in the set is assigned to one piece of the pie.
The pie chart automatically computes and displays the
percentage of the pie occupied by each data entry. Labels
can also be assigned to each segment of the pie chart as

well as adding titles to the graph. The pie chart shows the relationship of each data entry to the entire set of data.

The bar and stacked bar chart are similar to a histogram which can display six data ranges. To be most effective, the data ranges should have similar units. 1-2-3 automatically adjusts the vertical scale to accommodate the largest value being displayed. The bar chart gets more complicated because legends can be assigned to each of the data ranges, the X-axis can be labeled, the Y-axis can be labeled, and a vertical or horizontal grid can be displayed with the data. The user can also set the vertical scale to a fixed maximum value if desired. If a color monitor is available, the graph will display each data range in a separate color. When displayed on a monochrome monitor, each data range has a separate cross-hatched pattern.

A line graph can be used to display the same type of data as the bar chart. In the line graph, the user can elect to display the data with data symbols at each data point, with lines connecting the data points, or with both data symbols and lines. When symbols are used, 1-2-3 has a separate symbol for each data range. Each line and its associated symbols will be a different color if the color option is selected.

The XY graph is the only 1-2-3 graph which can display two dimensional data. It is the only graph which assigns a numerical scale to the x-axis as well as the vertical. Each data point represents the intersection of a pair of X and Y values. Like the line graph, the points can be represented by lines, symbols, or both.

```
     A  B  C  .  .  .     M  N  O  .  .  .          AA AB AC  .  .  .
 1                        |                   |
 2                        |                   |
 3                        |  DATA SUMMARY TABLE |
 4                        |                   |
 5                        |                   |               MACROS
 6                        |                   |
 7                        |---------------------|
 8      GPSS DATA         |                   |
 9                        |                   |
10                        |                   |
11                        |    RANGE NAMES    |
12                        |                   |
 .                        |                   |
 .                        |                   |
 .
```
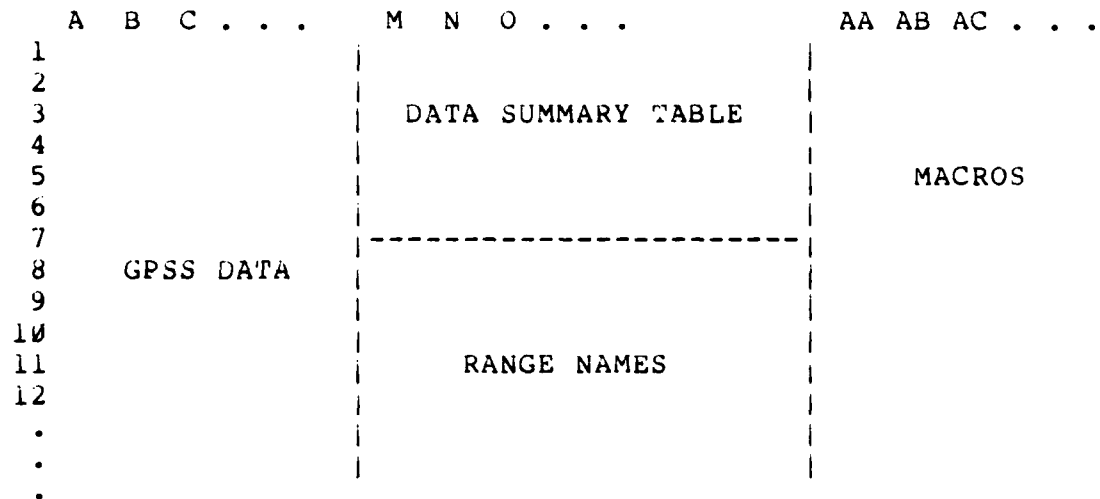
Figure 5. Spreadsheet organization scheme

Before getting into the actual procedures used to develop the graphics, it is helpful to consider the overall layout of the final 1-2-3 spreadsheet. The spreadsheet is really a large piece of electronic paper which can be divided into areas. It aids the organization and later use of the system to reserve specific areas of the spreadsheet for simulation program data, tables, macro commands, and

others.   The organization scheme used in this investigation
is illustrated in Figure 5.

A number of steps are required to get the simulation
data into the necessary format for graphing. First, the
.DIF files must be translated into Lotus 1-2-3 worksheet
format files. This is done to each of the five summary
report .DIF files with the translation utility program
provided by 1-2-3.

Second, the data from the five simulation summary
reports were combined on a single worksheet file. Lotus
1-2-3 offers three ways to combine worksheet files. They
can be added together, subtracted, or just copied to a
specified location on the worksheet. In this instance, the
File Combine Copy command was used to get the five summary
reports on the same worksheet in ascending order. The
combined simulation data worksheet was then saved in a
separate data file for later inclusion on the graphic macro
command worksheet.

The keyboard macro is a stored sequence of spreadsheet
commands activated by a single keystroke (15). Macros can
be used to simplify repetitive tasks which require many
keystrokes or to cut down the amount of time needed to

complete an operation. Macros execute commands much faster
than they could ordinarily be typed. A macro can also be
developed so a nontechnical user can perform complicated
functions with minimal instruction.

In this application, two sets of macros were used for
each set of graphs. The first macro generated a set of
graphs from the simulation data and assigned a name to each
graph. The second macro was used to recall the named graphs
in sequence for viewing. Only one macro was planned to
generate and display the graphs in sequence without
assigning each a name. However, with only one macro the
time between successive graphs was long and a number of
other pictures of the spreadsheet data flashed on the
monitor screen. The length of time and the flashing screens
were distracting to observers. When the two macro system
was used the time between successive graphs was almost cut
in half.

Each of the macros has some similarities. First, each
begins by resetting all of the graph settings in the
spreadsheet. This is like starting out with a clean sheet
of paper before constructing a new series of graphs.
Second, each uses range names to address the data rather
than cell locations. Ordinarily, 1-2-3 will change the cell

addresses in a formula to compensate for tne formula being moved to another location. Range names were used to allow the flexibility of moving tne macros on the spreadsheet as well as being easier to read than a cell address. Since 1-2-3 will remember tne range names but not display in a group, it was helpful to display them on a portion of tne spreadsneet for easy reference. Tne range names are snown in Figure 6.

MACRO Range Names

Name	Range	Name	Range	Name	Range
Q1M1	O6	Q1A1	Q6	INS1	U6
Q1M2	O6.O7	Q1A2	Q6.Q7	INS2	U6.U7
Q1M3	O6.O8	Q1A3	Q6.Q8	INS3	U6.U8
Q1M4	O6.O9	Q1A4	Q6.Q9	INS4	U6.U9
Q1M	O6.O10	Q1A	Q6.Q10	INS	U6.U10
Q2M1	R6	Q2A1	T6	ADJ1	V6
Q2M2	R6.R7	Q2A2	T6.T7	ADJ2	V6.V7
Q2M3	R6.R8	Q2A3	T6.T8	ADJ3	V6.V8
Q2M4	R6.R9	Q2A4	T6.T9	ADJ4	V6.V9
Q2M	R6.R10	Q2A	T6.T10	ADJ	V6.V10
FLD1	AY5.AY89	FLX	AW5.AW89	DUM	W6.W10
FLD2	AZ5.AZ89	FLY	AX5.AX89	DAY	N6.N10
FLD3	BA5.BA89				
FLD4	BB5.BB89				
FLD5	BC5.BC89				

Figure 6. Range name references

The data from the simulation cannot be accessed directly by tne graphing macros. The data to be graphed must first be copied to another location where similar data from each run appear in adjacent cells on tne spreadsheet.

In tnis case the data was copied into the table format in
Table 1. The range names from Figure 6 address tne cell in
the table. The graphs, in turn, address day 1 data, then
day 1 plus day 2, tnen day 1 plus day 2 plus day 3, and so
on.

Another common feature of all of the macros, like any
good program, is sufficient documentation. Figure 7 snows
tne macro used to generate tne queuing statistics graphics
with tne associated documentation. The eries of single
keystrokes wnicn makeup tne macro become rather cryptic with
tne passage of time. They are the first letters of 1-2-3
commands. The documentation in tne next column gives a
snort explanation of each macro line for later reference.

Table 1. Consolidated data table

			Data Summary Table				
	INSP LINE			ADJST LINE		INSP	ADJST
Days	MAX	CUR	AVE	MAX	CUR	AVE	UTIL	UTIL
Day 1	3	0	0.74	4	3	1.57	0.936	0.898
Day 2	3	0	0.70	6	4	3.01	0.952	0.949
Day 3	4	3	0.77	9	7	4.27	0.954	0.966
Day 4	4	0	0.84	12	10	5.52	0.955	0.974
Day 5	4	0	0.87	12	7	6.01	0.954	0.979

All of the graphics generation macros generate a series
of five graphs which display tne data summaries for each of

the five days sequentially. The first graph displays the
summary statistics for the first day and the second graph
displays both the first and second day summaries, and so on.
Each day represents a single set of data to be graphed. The
horizontal scale depicts days and data from at least two
days was needed because 1-2-3 would not graph just a single
set of data. This problem was overcome with the use of a
dummy data range. The dummy range was called "DUM" and
represented a range of five blank cells. This was enough to
make the 1-2-3 graphing function "see" five sets of data and
display the set of data for the first day.

The final feature each macro has in common is the use
of fixed scales. Ordinarily, 1-2-3 adjusts the vertical
scale to equal the maximum value to be displayed. This was
distracting to those viewing the graphs initially because
the vertical scale changed with each new graph in the
series. A fixed vertical scale set to the maximum value to
be displayed for the entire series was used later which
allowed viewers to focus on the data being displayed and not
on the changing scales.

The queuing macro in Figure 7 generates five successive
bar charts which display the maximum and average values for
the inspection and adjustment stations. The inspection data

are displayed in the A and B ranges and the adjustment data

are displayed in the D and E ranges.   The C range contains

the dummy data.

Commands	Description
/GRGTBAQ1M1~	Bar Graph, A=Q1Max
BQ1A1~CDUM~	B=Q1Ave, C=Dumb
DQ2M1~EQ2A1~	D=Q2Max, E=Q2Ave
XDAY~	X-axis
OLAInsp Que Max~	Label A Legend
LBInsp Que Ave~	Label B Legend
LDAdjt Que Max~	Label D Legend
LEAdjt Que Ave~	Label E Legend
TFTV INSPECTION AND REPAIR~	First Title
TSQueue Data~	Second Title
TXDAYS~	Label X-axis
TY# Waiting~	Label Y-axis
SYL~U12~MQ	Manual Y-axis Scale
CGHQ	Color, Horizontal Grid
VNCQUEU1~	View, Name Graph
AQ1M2~BQ1A2~	Add Second Day Data
DQ2M2~EQ2A2~	
VNCQUEU2~	View, Name Graph
AQ1M3~BQ1A3~	Add Third Day Data
DQ2M3~EQ2A3~	
VNCQUEU3~	View, Name Graph
AQ1M4~BQ1A4~	Add Fourth Day Data
DQ2M4~EQ2A4~	
VNCQUEU4~	View, Name Graph
AQ1M~BQ1A~	Add Fifth Day Data
DQ2M~EQ2A~	
VNCQUEUE~	View, Name Graph

Figure 7. Macro to generate queue graphs

Once the graphs are generated and checked by viewing,

they are named by the macro before going on to the next set

of data. Naming the graph saves the graph settings for that

particular graph and makes it possible to use the shorter

macro in Figure 8 to recall the graph. Adding this extra

step cut the time between displayed graphs almost in half.
The graph leaves the screen for approximately three seconds
while 1-2-3 recalls the settings for the next graph.

Command	Description
/GNUQUEU1~	Use Graph QUEU1
NUQUEU2~	Use Graph QUEU2
NUQUEU3~	Use Graph QUEU3
NUQUEU4~	Use Graph QUEU4
NUQUEU~	Use Graph QUEU

Figure 8.   Macro to display queue graphs

Command	Description
/GRGTLAINS1~	Line Graph, A=Insp
BDUM~CADJ1~	B=Dum, C=Adj
XDAY~	X-axis
OLAInspectors~	Label A Legend
LCAdjuster~	Label C Legend
TFTV INSPECTION AND REPAIR~	First Title
TSStation Utilization Rates~	Second Title
TXDays~	Label X-axis
TY% Busy~	Label Y-axis
SYL.89~U1.0~MQ	Fix Y-axis Scale
CGHQ	Set Color, Horizontal Grid
VNCSERV1~	View, Name Graph
AINS2~CADJ2~	Second Day Data
VNCSERV2~	View, Name Graph
AINS3~CADJ3~	Third Day Data
VNCSERV3~	View, Name Graph
AINS4~CADJ4~	Fourth Day Data
VNCSERV4~	View, Name Graph
AINS~CADJ~	Fifth Day Data
VNCSERV5~	View, Name Graph

Figure 9.   Macro to generate utilization line graphs

The utilization rates of the inspectors and the
adjuster are compared in a series of graphs generated by the
macro in Figure 9.   This macro uses a line graph with both

symbols and lines connecting the symbols to display the
data.

Command	Description
/GRGTXXFLX~	XY Graph, Set X Range
AFLY~	Set A Range
OSYL~U50~MQ	Manual Y-axis
SXL~U50~MQ	Manual X-axis
FGLQ	Lines Only
TFTV INSPECTION AND REPAIR~	First Title
TSFlow Representation~	Second Title
DAFLD1~CQQ	First Day Data
VNCSCH1~	View, Name Graph
ODAFLD2~CQQ	Second Day Data
VNCSCH2~	View, Name Graph
ODAFLD3~CQQ	Third Day Data
VNCSCH3~	View, Name Graph
ODAFLD4~CQQ	Fourth Day Data
VNCSCH4~	View, Name Graph
ODAFLD5~CQQ	Fifth Day Data
VNCSCH5~	View, Name Graph

Figure 10. Macro to generate schematic diagrams

The queuing macro and utilization graphs each display
like quantities. The macro in Figure 10 uses an XY graph to
display queue and utilization data together. This macro
uses the information from Figure 11 to draw the schematic
diagrams of the inspection and adjustment operation seen in
Figures 22 through 26. The utilization rates and the
current number of TV sets waiting at each station are also
displayed at the appropriate location on the graph. The X
and Y coordinate numbers were selected to achieve a
proportional representation of the operation based on the
maximum X and Y scale values.   1-2-3 views these numbers as

data points and connects those in adjacent cells with a
line. Separate figures or lines are achieved by the blank
lines between sets of numbers. The data and labels on the
diagram are placed there by using the data-label option with
the selected data points.

The macros were developed after the simulation data
were generated, translated to Lotus 1-2-3 worksheet format,
and combined on a single spreadsheet. Subsequent simulation
runs of this problem require the simulation standard reports
to be translated and combined. This combined simulation
report can then be copied over the first run data and the
relevant data copied to the consolidated data table. This
spreadsheet can be saved in a separate file and the process
repeated for more runs if necessary.

	X-axis	Y-axis
Waiting	7	15
Insp	7	20
	17	20
	17	15
	7	15
Insp#1	21	19
	21	23
	24	23
	24	19
	21	19
Insp#2	21	16
	24	16
	24	12
	21	12
	21	16
Waiting	27	32.5
Adjust	27	37.5
	37	37.5
	37	32.5
	27	32.5
Adjust	40	32.5
	40	37.5
	43	37.5
	43	32.5
	40	32.5
Arrival	2	17.5
Line	6.5	17.5
	18	17.5
	20.5	21
	17.5	17.5
	18	17.5
	20.5	14
	24.5	21
	29	17.5
	24.5	14
	29	17.5

	X-axis	Y-axis		
Depart	29	17.5		
	35	17.5		
Insp to	22.5	24		
Adjust	22.5	35		
	26.5	35		
Wait to	37.5	35		
Adjust	39.5	35		
Return	43.5	35		
to Insp	47	35		
	47	42		
	12	42		
	12	21		
Labels	4	20	Arrivals	
	40	17.5	To Shipping	
	22.5	10	Inspection	
	12	13	# Waiting	
	32	30.5	# Waiting	
	25	44	Return for Inspection	
	41.5	30.5	Adjustment	
Data				
Adj Ut	41.5	28	0.898	
Insp Ut	22.5	7	0.936	
Wt Insp	12	17.5	0	
Wt Adj	32	35	3	
	35	10	Day 1	

Figure 11. Data for schematic

RESULTS AND DISCUSSION

The figures in this section illustrate the results of using Lotus 1-2-3 to display GPSS/PC simulation data. Once the simulation data were consolidated in the form seen in Table 1 above, the 1-2-3 macros generated these graphs with a single keystroke. When generated, they can be called up on the microcomputer monitor at any time with a single keystroke.

A number of steps were required to get the data into the format of Table 1. The model in Figure 1 was run with the control statement sequence from Figure 2. This generated five unformatted report files such as the one in Appendix B. These five files were formatted and saved using GPSSREPT, the GPSS report processing program. The formatted reports were then translated to .DIF files using the DIFFER program supplied by GPSS.

The .DIF files were then translated to the Lotus 1-2-3 .WKS format using the 1-2-3 translation utility. The five .WKS files were combined on a single spreadsheet using the 1-2-3 File Combine Copy command with the cursor placed in a cell immediately below the previous summary report. This single file was saved and later combined onto the macro

spreadsheet using the same File Combine Copy command. The important step here is that the combined simulation data must be copied to the macro spreadsheet because tne macro spreadsheet contains all of the named command ranges from Figure 6. If the macro spreadsneet is copied to the simulation data spreadsheet, all of the range names will have to be renamed using the Range Name Create command. Tne macros and the data in Figure 6 will be combined on the simulation data spreadsheet, but the command ranges themselves are not transferred.

With the five simulation runs on the macro worksheet, the selected data to be graphed are copied to the appropriate cell of Table 1. As can be seen by a comparison of the summary report in Appendix A and the data in Table 1, not all of the simulation data were displayed graphically. This is a decision of the modeler based on tne specific areas of interest in the simulation analysis. In tnis case, data on average service times at each station, number of entries at each station or queue, or average time in each waiting line were not used.

The graphs are generated by invoking tne appropriate range name for tne individual macro. In this case, the queue graphs were generated and saved by the ALT Q command,

where Q is the range name given to the queuing macro. The
server utilization graphs were generated by ALT S and the
schematic diagrams by ALT F. The generated queue graphs can
be recalled by ALT A, the server utilizations by ALT B, and
the schematic diagrams by ALT P.

The data were separated into three categories for
display. The categories were selected to snow how the data
could be displayed in graph form. Queue statistics,
maximums and averages, were compared with queue statistics
in Figures 12, 13, 14, 15, and 16, because 1-2-3 can only
display one set of units on the ordinate axis. The bar
chart was chosen to display the queue statistics because it
is one of the graphs able to display up to six separate data
ranges.

In this example, the data were taken at the end of each
day. The display then gives a series of "snapshots" of data
for each day. The series of snapshots is begun with a
single keystroke and halts at each snapshot for viewing the
graph. The macro advances to the next snapshot when the
RETURN key is depressed.

Figures 12 through 16 clearly snow the relationship of
the queue data. The graph makes it easy to see that for

this simulation run, the inspection queue reaches its
maximum value by the third day while the average remains
constant. The adjustment queue continues to increase
reaching an apparent maximum value on the fourth day while
the average continues to increase over the entire five day
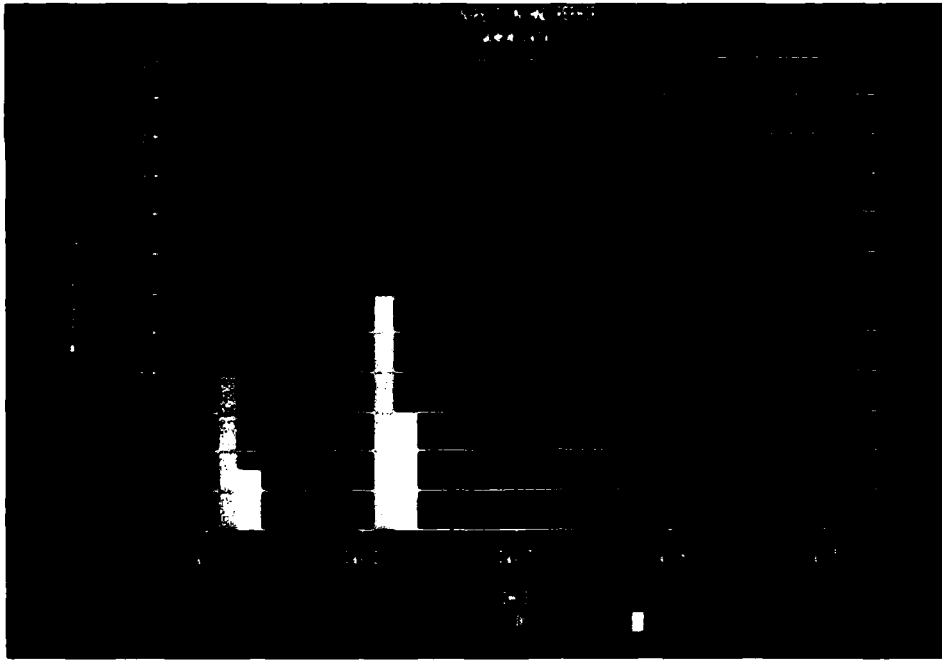period.
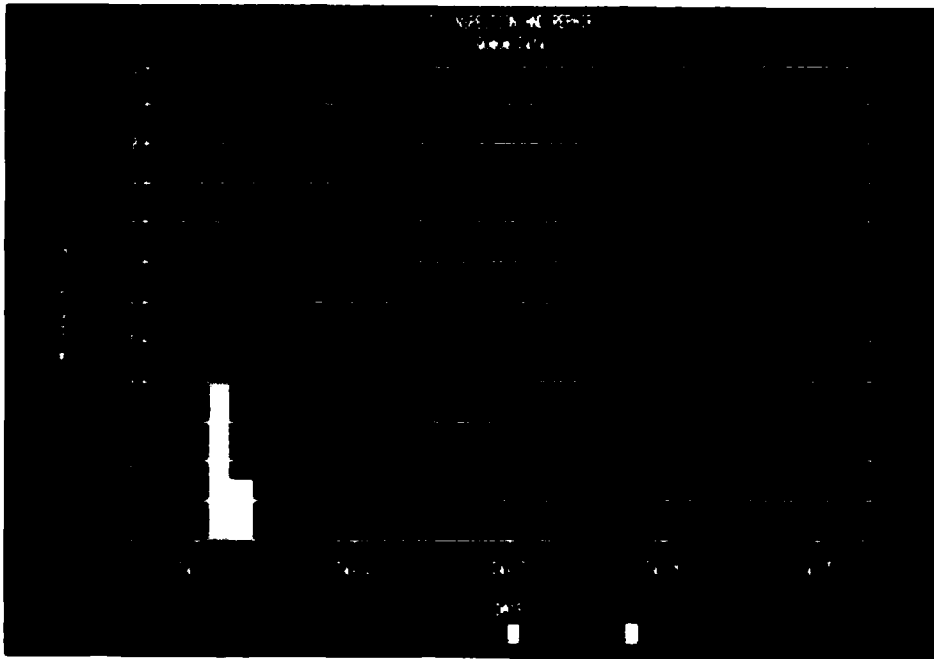
Figure 12. Queue data for the first day

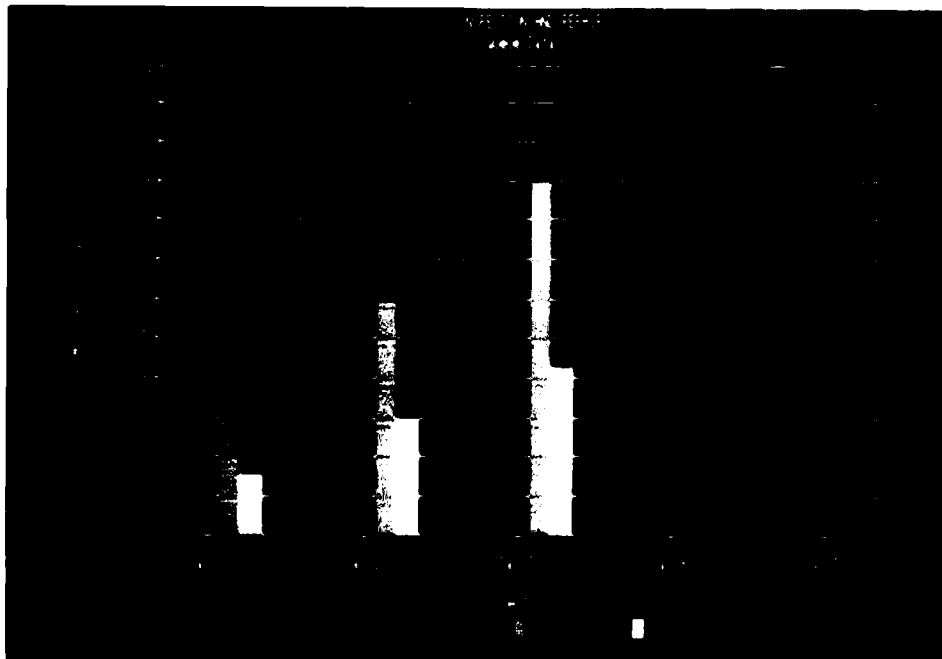Figure 13.   Queue data for the second day
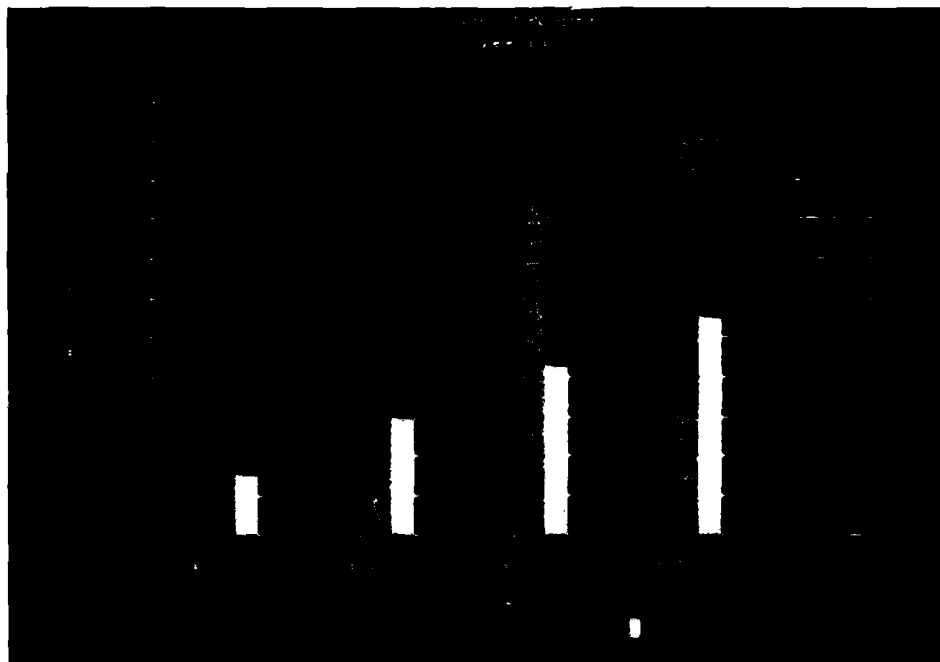

Figure 14.   Queue data for the third day
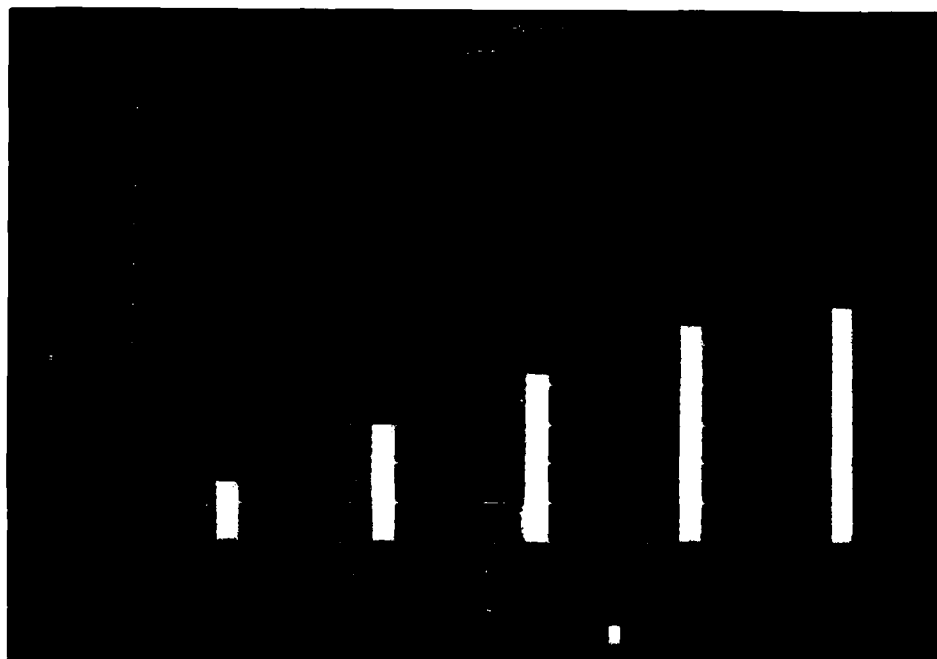
43

Figure 15. Queue data for the fourth day

Figure 16. Queue data for the fifth day

A line graph was chosen to illustrate the utilization
rates of the inspectors and the adjuster. The line graph
displays a symbol for each simulation data point and
connects the data points with a line. The line graph was
chosen in this instance because it best illustrates the
trends occurring in the utilization rates. The macro in
Figure 9 generated the utilization graphs in Figures 17
through 21. Figure 17 displays only a single symbol for
each utilization on the ordinate axis since it represents
the first day. The graphs clearly snow that the inspector
utilization remains relatively constant on days two through
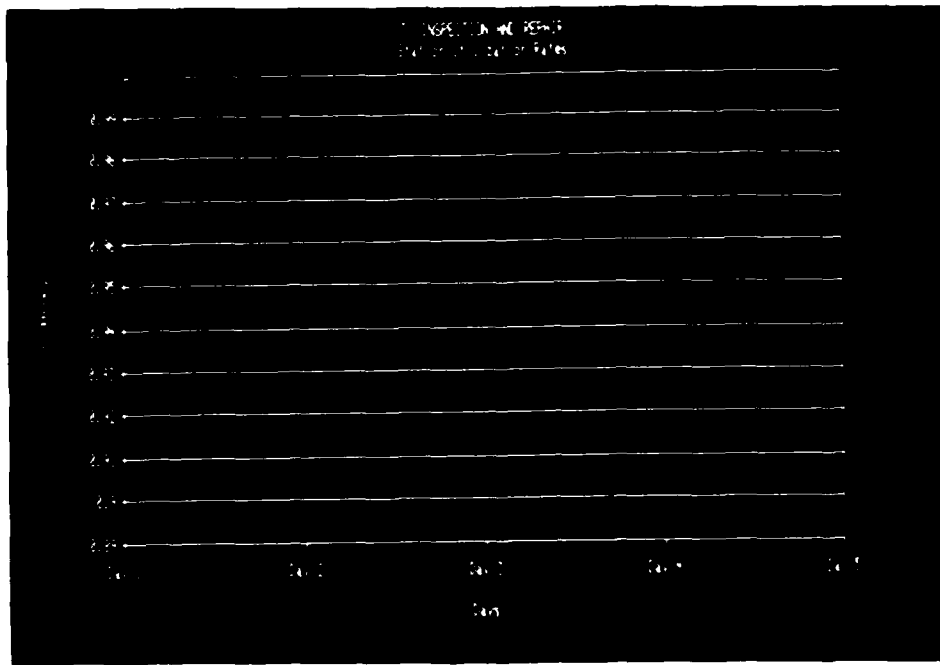five while the adjuster utilization continues to increase
during the five days.

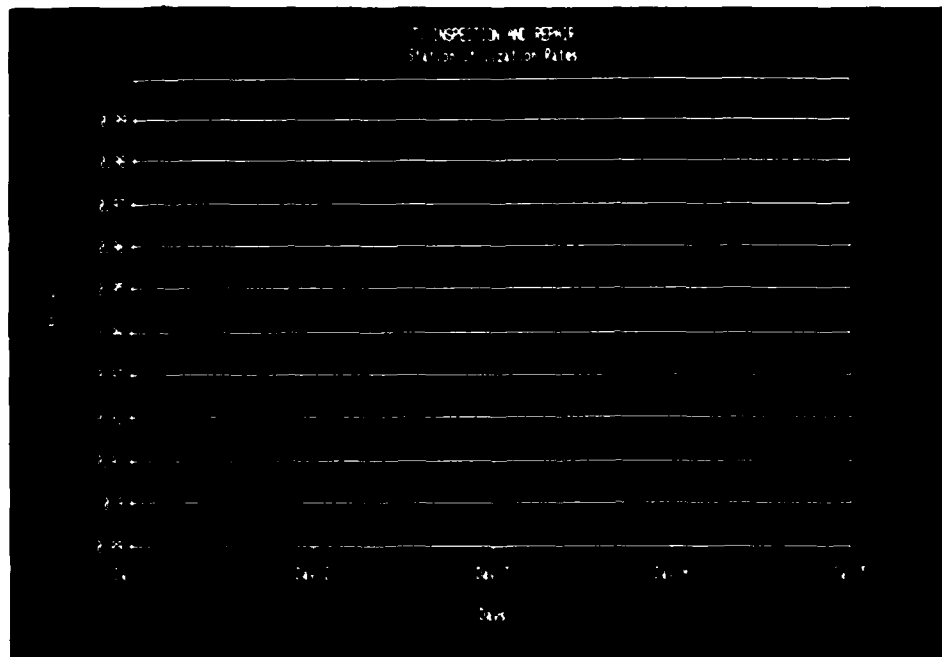Figure 17. Utilization rates for the first day

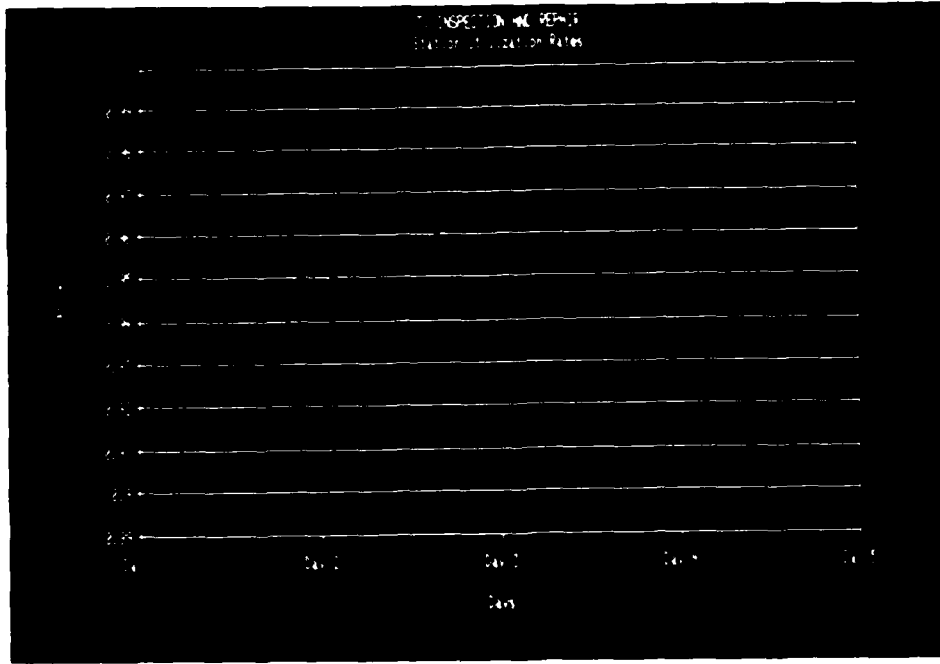Figure 18. Utilization rates for the second day

46

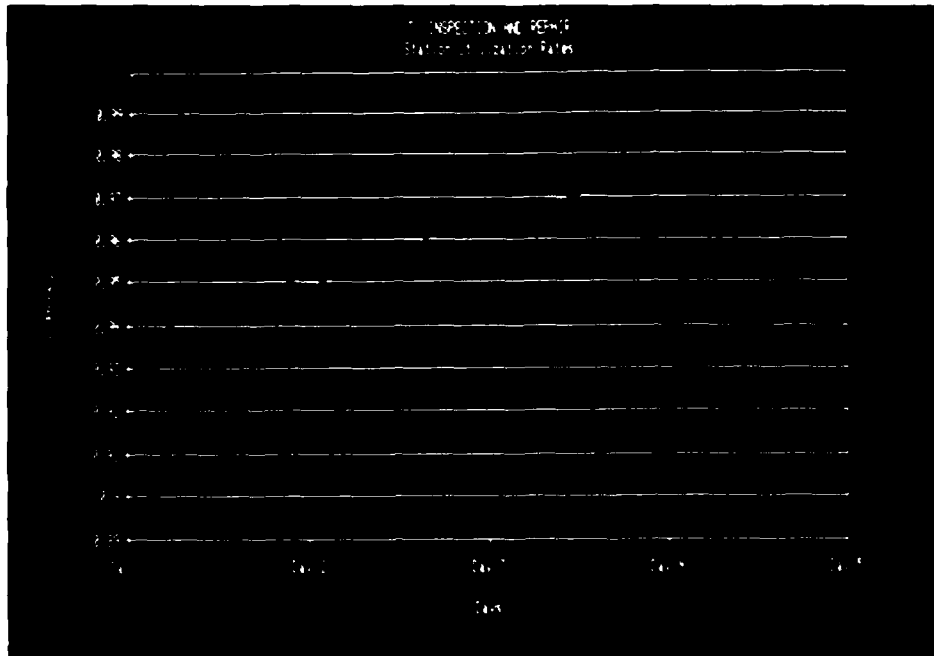Figure 19.   Utilization rates for the third day

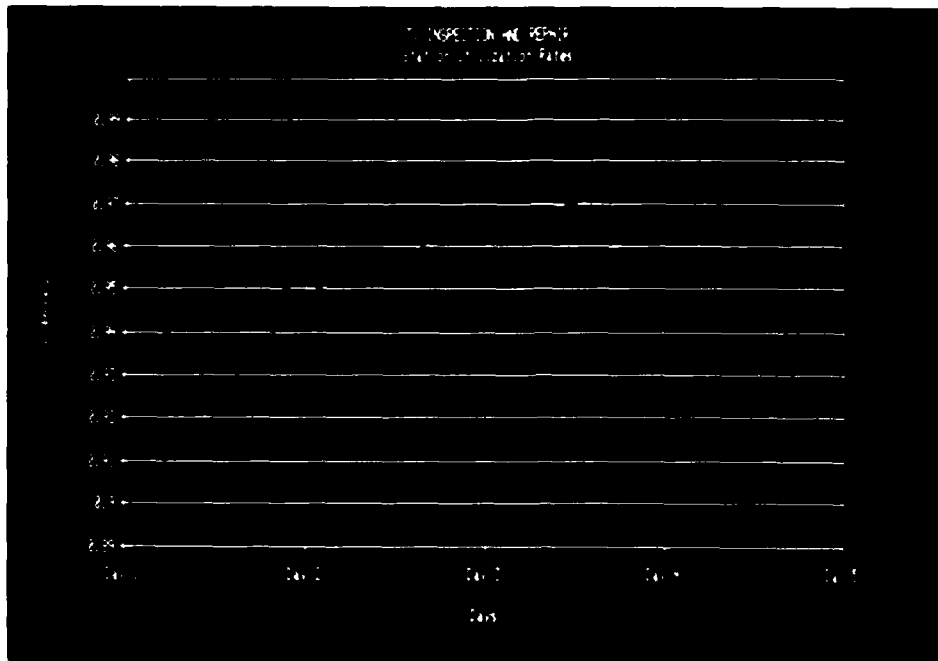Figure 20.   Utilization rates for the fourth day

Figure 21.   Utilization rates for the fifth day

While comparing utilization rates to utilization rates
and queue statistics to queue statistics is all well and
good, it does not tell the whole story.   It is also of
interest to look at both the queue statistics and the
utilization rates at the same time.   1-2-3 has difficulty
displaying dissimilar quantities such as this on the bar
chart or line graph.   Displaying these quantities may also
be a bit abstract for those not directly involved in the
model development.

The solution to this problem was to develop a schematic
flow diagram of the inspection and adjustment operation

which included data from the simulation. The graphs in
Figures 22 through 26 were developed by the data in Figure
11. Queue and utilization data from the simulation run were
added to the figures using a 1-2-3 data labeling technique.
The macro in Figure 10 was used to generate a series of
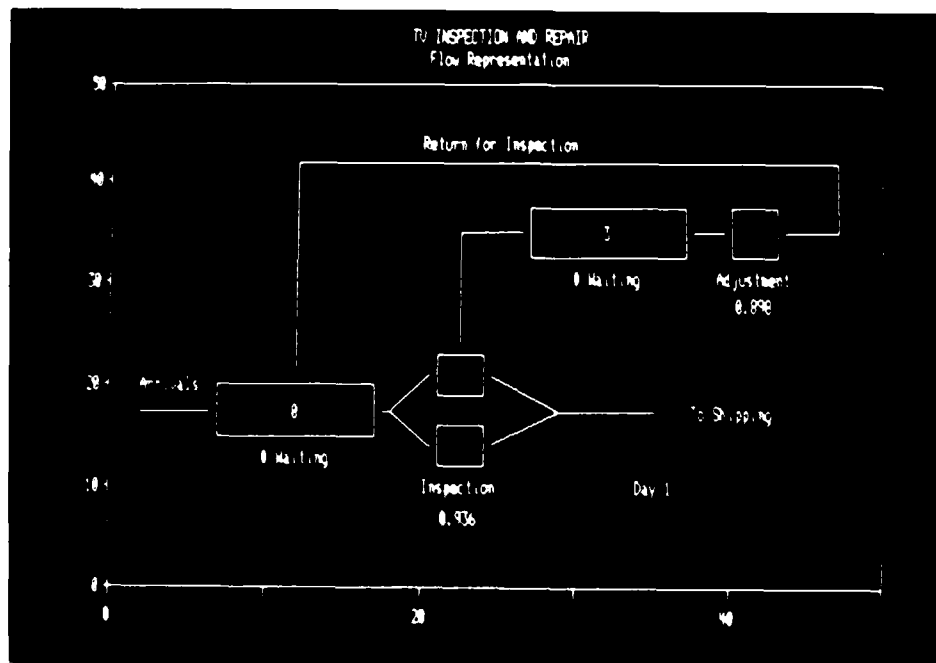schematic diagrams to display the data summary for each of
the five days.

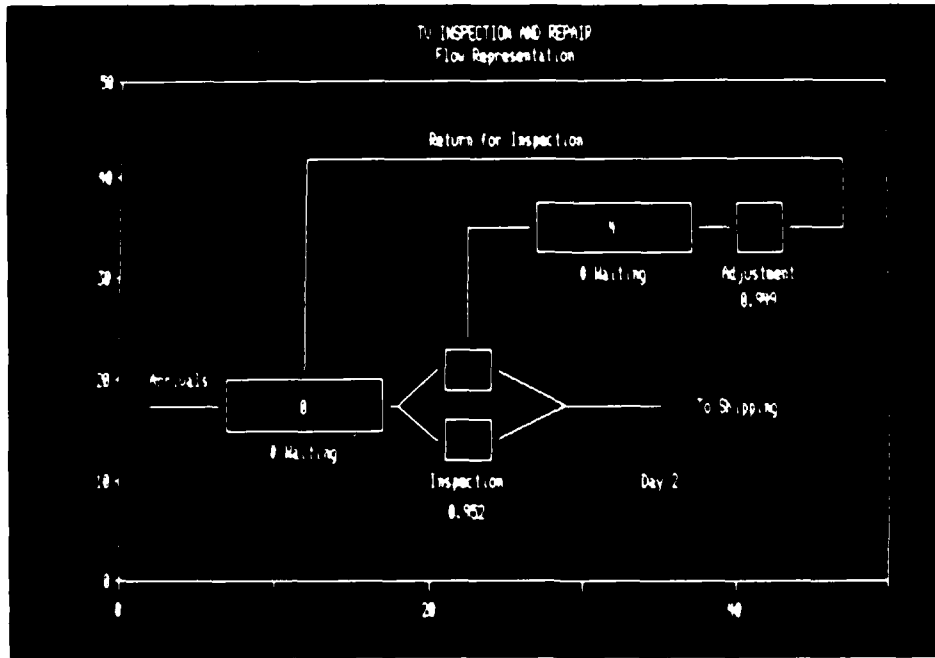Figure 22. Schematic diagram for the first day
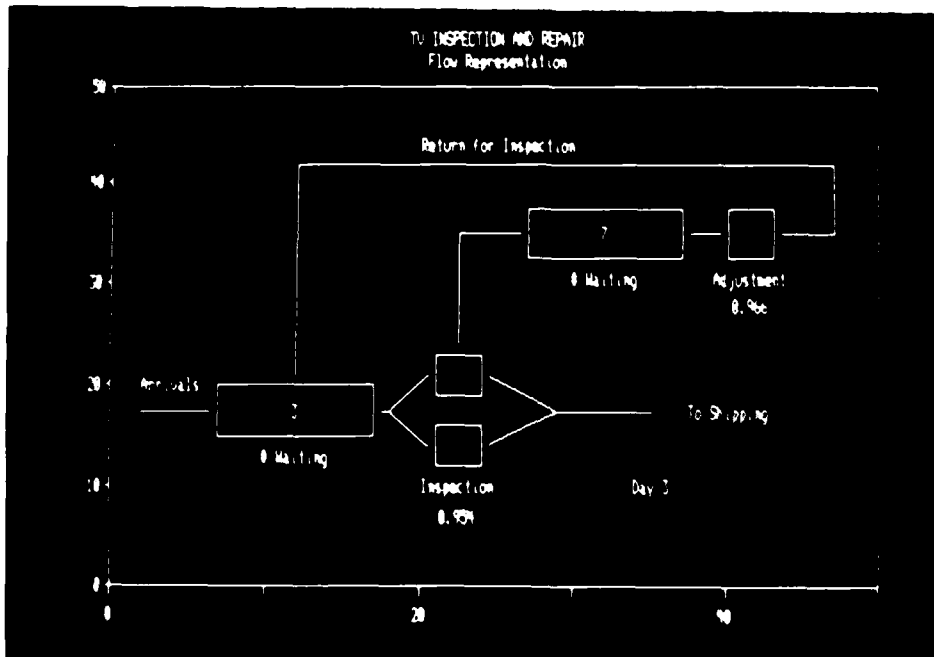
Figure 23.   Schematic diagram for the second day

Figure 24.   Schematic diagram for the third day

50
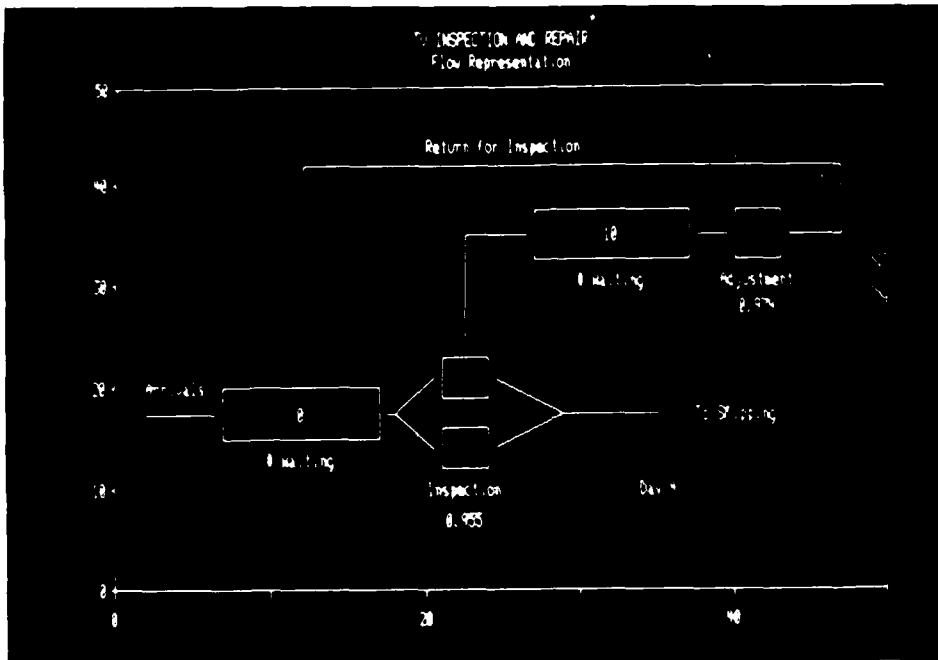
Figure 25.   Schematic diagram for tne fourth day
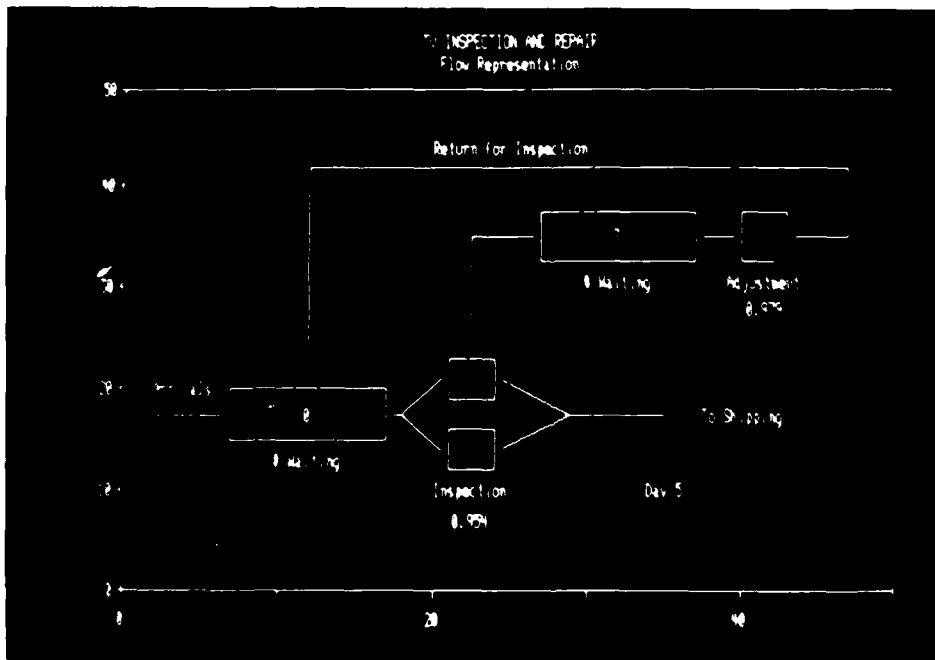
Figure 26.   Schematic diagram for the fifth day

Finally, the model residence time table of the SNA M1 was graphed manually to illustrate a possible use of the pie chart. 1-2-3 automatically assigned a percentage figure to each segment of the pie and the data labels were modified from the simulation table. The image in Figure 27 appears as an oval because of the aspect ratio used by 1-2-3 to draw the circle. The pie would appear more circular if sent to a printer or plotter supported directly by 1-2-3 because 1-2-3 would then use the correct aspect ratio for the specific output device.
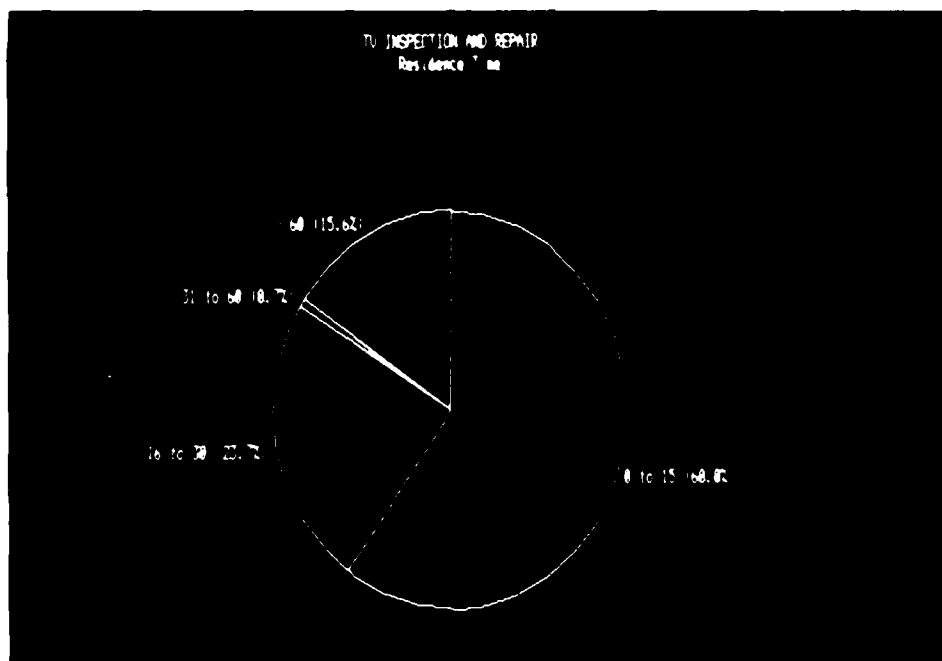
Figure 27. Inspection/adjustment throughput time

While this investigation focused on interfacing GPSS/PC with Lotus 1-2-3, the similar steps can also be done with other simulation software using .DIF. Data from the microcomputer version of SLAM II were also translated into a .WKS file from a SLAM II generated .DIF file. The SLAM MONTR statement was used to generate summary reports at the desired time intervals. Rather than go through a two step process from unformatted to formatted to .DIF file, SLAM gives the user a choice of a formatted report or a .DIF file output. The latter converts the unformatted summary file directly to a .DIF file. When this .DIF file is translated to a .WKS file it has the same data and format as the standard formatted report.

File	Memory Required
REPORT.GPS	635 Bytes
BARSHOP.DIF	15875 Bytes
BARSHOP.WKS	5305 Bytes

Figure 28. GPSS/PC disk memory requirements

File	Memory Required	
BARBR.TRA	91019 Bytes	input translation
BARBR.OUT	91019 Bytes	execution
BARBR	898 Bytes	output
BARBR.DIF	4269 Bytes	
BARBR.WKS	1479 Bytes	

Figure 29. SLAM II disk memory requirements

Other differences were also noted between GPSS/PC and SLAM II which have some bearing on the usefulness of this technique on certain hardware configurations. SLAM has much greater intermediate disk memory requirements which became unwieldy with a number of intermediate data summaries. A comparison was made with a single server model for the same simulated time period which generated the files and data storage requirements of Figures 28 and 29. The same single server barbershop model on SLAM required substantially more disk memory requirements than GPSS/PC.

The large intermediate disk memory requirements make SLAM cumbersome to use on a microcomputer without a hard disk. GPSS/PC is able to perform the simulation directly from the source code without compiling an executable file like SLAM. Even with the two-step conversion from unformatted to formatted to .DIF file GPSS/PC uses less disk memory than SLAM II to accomplish the same simulation.

54

CONCLUSIONS

The techniques developed in this investigation show now
Lotus 1-2-3 can be used as an effective means to graphically
display simulation output. Although, pure animation was not
possible, the semi-animated "snapshot" method developed may
actually be more useful in certain situations.

The snapshot method allows the analyst to focus
management's attention on those times when significant
critical conditions occur. The analyst is free to choose
the desired time increment and could, theoretically, reach a
nearly animated condition if the increment were small
enough. Use of the GPSS/PC PLOT statement can assist the
analyst in determining critical time periods in the
simulation. Frequent data summaries taken at intervals
around the critical time periods could then be graphed on
Lotus 1-2-3.

Lotus 1-2-3 macros make it easy for the analyst to
develop a set of standard graphs which can be used over and
over on similar simulation studies. A set of graphing
macros could be developed for each of a series of problems
being studied. Once the initial macros are developed it is
relatively easy to adapt them to new problems with

additional queues or servers.

True animation was not possible because GPSS/PC does not have the statements to interface with storage devices and subroutines found in the main frame version of the language. GPSS/PC is only capable of writing an unformatted output file at the end of a simulation run using the standard output supplemented by user requested tables of SNA values or SAVEVALUES. This not only prevents the use of animated graphics, but may also limit the analyst's ability to develop a model which accurately reflects a unique situation outside the capability of GPSS/PC block statements.

All simulation projects require the involvement of management at some point. The more clearly management understands the benefits of the simulation results, the better chance necessary changes will be adopted. In other words, simulation analysts must sell the simulation results to the decision makers. Graphics are one method to enhance the selling of the simulation study. Without good salesmanship, the results of the best simulation study may not be understood or adopted by management.

The techniques developed in this study enable the

simulation modeler to get graphic output from a simulation model with minimum programming effort and cost. Lotus 1-2-3 offers the modeler the flexibility to choose from a number of different graphs and charts to present the results in a format best suited to the particular data. With some additional effort, the modeler can use 1-2-3 to draw flow diagrams or any other type of graphic the modeler requires. Use of Lotus 1-2-3 in conjuction with GPSS/PC makes it easy for the modeler to present simulation results in a readable and interesting format.

Further research could be done on this technique to develop a more generalized framework for graphic display of simulation data. Perhaps a system could be developed which would not require major changes to the display generation macros for each new simulation model. Another area for further study would be to assess the actual benefits of displaying simulation output data graphically versus the standard output. Additional work could be done to investigate the improvements made in Lotus 1-2-3 version 2. Regression analysis and other features have been added which could be of use in analyzing pre-simulation data from the workplace.

BIBLIOGRAPHY

1. Banks, J. and Carson, J. S., II. "Process-interaction Simulation Languages." Simulation 44 (May, 1985): 225-235.

2. Carroll, T. O. and Carroll, D. W. "Network Analysis with Spreadsheets." Lotus 1 (December, 1985): 70-74.

3. CINEMA. State College, PA: Systems Modeling Corporation, 1985.

4. DIF Technical Specification. Newton Lower Falls, MA: Software Arts Products Corporation, 1983.

5. Favreau, R. R.; Murtha, S. A. and Marr, G. R., Jr. "Desk-top Simulation - A State-of-the-Art Perspective." The Proceedings of the 1984 Summer Computer Simulation Conference, pp. 217-224. La Jolla, CA: Society for Computer Simulation, 1984.

6. Feltner, C. E. and Weiner, S. A. "Models, Myths and Mysteries in Manufacturing." Industrial Engineering 17 (July, 1985): 66-76.

7. Friedman, R. "Ideas and Trends." Lotus 1 (December, 1985): 24.

8. GPSS/PC User Manual. Stowe, MA: Minuteman Software, 1985.

9. Haynes, J. L. "Circuit Design with Lotus 1-2-3." Byte 10 (Special IBM Issue, 1985): 143-156.

10. Kachitvichyanukul, V. and Lloyd, C. M. C. "A Microcomputer-based Integrated Simulation Support (MISS) System." The Proceedings of the 1985 Summer Computer Simulation Conference, pp. 9-13. San Diego: Society for Computer Simulation, 1985.

11. Kaplan, S. J. "Some Future Trends in Spreadsheets." Lotus 2 (February, 1986): 138.

12. Karian, Z. A. "GPSS/PC." Byte 10 (October, 1985): 295-301.

13. Lashway, S. and Pendergast, N. "The New 1-2-3." Lotus 1 (September, 1985): 60-64.

14. LeBlond, G. T. and Cobb, D. F. Using 1-2-3.
    Indianapolis: Que Corporation, 1983.

15. Lotus 1-2-3 User's Manual. Cambridge, MA: Lotus
    Development Corporation, 1983.

16. Macintosh, J. B.; Hawkins, R. W. and Shepard, C. J.
    "Simulation on Microcomputers: The Development of a
    Visual Interactive Modeling Philosophy." Proceedings
    of the 1984 Winter Simulation Conference, pp. 531-536.
    Piscataway, NJ: Institute of Electrical and Electronic
    Engineers, Inc., 1984.

17. Masri, S. and Moodie, C. "Using an Electronic
    Spreadsheet to Analyze Manufacturing Flow Systems."
    Computers and Industrial Engineering 9 (2) (1985):
    183-193.

18. McMillan, D. A. "Publisher's Letter." Lotus 1
    (December, 1985): 4.

19. Minor, R. J.; Sabuda, J. D. and Wortman, D. B.
    "Graphics and Simulation: Tools and Applications."
    Proceedings of the 1982 Winter Simulation Conference,
    pp. 161-171. Piscataway, NJ: Institute of Electrical
    and Electronic Engineers, Inc., 1982.

20. Morse, L. and Whitehouse, G. E. "Electronic
    Spreadsheets Evaluated for IE Applications."
    Industrial Engineering 17 (February, 1985): 17-26.

21. Morse, L. and Whitehouse, G. E. "IEs Must Look at
    Equipment, Needs in Choosing Spreadsheets." Industrial
    Engineering 17 (March, 1985): 21-31.

22. Mourant, R. R. "Turbosim – A Simulation Language for
    Microcomputers." The Proceedings of the 1985 Summer
    Computer Simulation Conference, pp. 26-31. San Diego:
    Society for Computer Simulation, 1985.

23. Parrinello, J. "Using the Personal Computer to Project
    Space Requirements." Computers and Industrial
    Engineering 9(4) (1985): 347-355.

24. Pritsker, A. A. B. Introduction to Simulation and SLAM
    II. Second edition. New York: Halsted Press, 1984.

25. Reese, R. M. "Simulation Programming Using Spreadsheet
    Software." Proceedings of the 1984 Winter Simulation
    Conference, pp. 539-542. Piscataway, NJ: Institute
    of Electrical and Electronic Engineers, Inc., 1984.

26. Schriber, T. J. Simulation Using GPSS. New York:
     John Wiley and Sons, 1974.

27. Standridge, C. R.; Hoffman, J. R. and Walker, S. A.
     "Presenting Simulation Results With TESS Graphics."
     Proceedings of the 1984 Winter Simulation Conference,
     pp. 305-311. Piscataway, NJ: Institute of Electrical
     and Electronic Engineers, Inc., 1984.

28. Vasudev, V. K. and Pruett, J. M. "Simulation on Three
     Sizes of Computers - An Evaluation." Computers and
     Industrial Engineering 9(3) (1985): 263-271.

29. Yancey, D. P. and Phillips, D. T. "Simulation Output
     Analysis within IDSS Prototype (2.0)." 1983 Annual
     Industrial Engineering Conference Proceedings, pp.
     471-479. Norcross, GA: Institute of Industrial
     Engineers, 1983.

ACKNOWLEDGEMENTS

61

APPENDIX A

GPSS/PC STANDARD FORMATTED REPORT

```
15                              79          0          0
16                               5          0          0
17                               5          0          0
```

	TRANSFER	79	0	0
	GENERATE	5	0	0
	TERMINATE	5	0	0

FACILITY	ENTRIES	UTIL.	AVE.TIME	AVAILABLE	OWNER	PEND	INTER	RETRY	DELAY
ADJT	80	0.979	293.90	1	386	0	0	0	7

QUEUE	MAX	CONT.	ENTRIES	ENTRIES(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
LINE	4	0	507	107	0.87	41.16	52.17	0
LIN2	12	7	87	2	6.01	1659.01	1698.05	0

STORAGE	CAP.	REMAIN.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
INSP	2	0	0	2	507	1	1.91	0.954	0	0

TABLE	MEAN	STD.DEV.	RETRY	RANGE		FREQUENCY	CUM. %
RTABLE	495.71	1008.96	0				
				0 -	150	251	60.05
				150 -	300	99	83.73
				450 -	600	3	84.45
				600 -		65	100.00

64

APPENDIX B

GPSS/PC STANDARD UNFORMATTED REPORT

```
;  GPSS/PC Unformatted Report file B:REPORT.GPS.
(V 1.1, # 30712)  02-12-1986 14:04:58
DAY 5 SUMMARY
  1  10009  0   24000   18   1   1   4848
  2  10001  2   0   0   0   0   0
RTABLE
  2  10002  2   0   0   0   0   0
INSP
  2  10003  2   0   0   0   0   0
LINE
  2   2   3   0   0   0   0   0
STA1
  2  10   3   0   0   0   0   0
STA2
  2  10006  2   0   0   0   0   0
LIN2
  2  10007  2   0   0   0   0   0
ADJT
  2  10008  2   0   0   0   0   0
B:DAY1.GPS
  2  10009  2   0   0   0   0   0
B:REPORT.GPS
  3   0   1   428   0   0   17   0
  3   0   2   507   0   0   32   0
  3   0   3   507   0   0   10   0
  3   0   4   507   0   0   9   0
  3   0   5   507   2   0   1   0
  3   0   6   505   0   0   22   0
  3   0   7   505   0   0   47   0
  3   0   8   418   0   0   43   0
  3   0   9   418   0   0   44   0
  3   0  10   87   7   0   32   0
  3   0  11   80   0   0   39   0
  3   0  12   80   0   0   9   0
  3   0  13   80   1   0   1   0
  3   0  14   79   0   0   33   0
  3   0  15   79   0   0   47   0
  3   0  16   5   0   0   17   0
  3   0  17   5   0   0   44   0
  3   0  20000   0   0   0   44   0
  4  10007  80   0   979   293.9   1   386
  5  10007   0   0   0   7   0   0
  6  10003   4   0   507   107   0   0
  7  0.8695   41.1597633   52.17   0   0   0   0
  6  10006  12   7   87   2   0   0
  7  6.0139167   1659.0114943   1698.0470588   0   0   0   0
```

```
8   10002   2   0   0   2   507   1
9   1.907625   953.8125   0   0   0   0   0
10   10001   5   0   150   150   495.7129187   1008.9565205
11   418   251   99   0   3   65   1
```

APPENDIX C

GPSS/PC REPORT IN .DIF FORMAT

```
TABLE
 0 , 1
""
VECTORS
 0 , 50
""
TUPLES
 0 , 10
""
DATA
 0 , 0
""
-1 , 0
BOT
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
"GPSS/PC"
 1 , 0
""
 1 , 0
"START_TIME"
 0 , 0
V
 1 , 0
""
 1 , 0
""
 1 , 0
"NAME"
 1 , 0
"INSP"
 1 , 0
"LINE"
 0 , 1
V
 0 , 2
V
 0 , 2
V
 1 , 0
```

```
"ADJT"
 1 , 0
"B:REPORT.GPS"
 1 , 0
""
 1 , 0
""
 1 , 0
"LOC"
 0 , 1
V
 0 , 1
V
 0 , 3
V
 0 , 4
V
 0 , 5
V
 0 , 6
V
 0 , 7
V
 0 , 8
V
 0 , 2
V
 0 , 10
V
 0 , 11
V
 0 , 12
V
 0 , 13
V
 0 , 14
V
 0 , 15
V
 0 , 16
V
 1 , 0
""
 1 , 0
""
 1 , 0
"FACILITY"
 1 , 0
"ADJT"
 1 , 0
```

```
""
 1 , 0
""
 1 , 0
"QUEUE"
 1 , 0
"LINE"
 0 , 2
V
 1 , 0
""
 1 , 0
""
 1 , 0
"STORAGE"
 1 , 0
"INSP"
-1 , 0
BOT
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
"Report"
 1 , 0
""
 1 , 0
"END_TIME"
 0 , 24000
V
 1 , 0
""
 1 , 0
""
 1 , 0
"VALUE"
 0 , 10001
V
 0 , 10002
V
 0 , 2
V
 0 , 9
V
 0 , 10005
```

```
V
 0 , 10006
V
 0 , 10007
V
 1 , 0
""
 1 , 0
""
 1 , 0
"BLOCK_TYPE"
 1 , 0
"GENERATE"
 1 , 0
"QUEUE"
 1 , 0
"ENTER"
 1 , 0
"DEPART"
 1 , 0
"ADVANCE"
 1 , 0
"LEAVE"
 1 , 0
"TRANSFER"
 1 , 0
"TERMINATE"
 1 , 0
"QUEUE"
 1 , 0
"SEIZE"
 1 , 0
"DEPART"
 1 , 0
"ADVANCE"
 1 , 0
"RELEASE"
 1 , 0
"TRANSFER"
 1 , 0
"GENERATE"
 1 , 0
"TERMINATE"
 1 , 0
""
 1 , 0
""
 1 , 0
"ENTRIES"
 0 , 73
```

```
V
 1 , 0
""
 1 , 0
""
 1 , 0
"MAX"
 0 , 6
V
 0 , 9
V
 1 , 0
""
 1 , 0
""
 1 , 0
"CAP."
 0 , 2
V
-1 , 0
BOT
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
"file"
 1 , 0
""
 1 , 0
"BLOCKS"
 0 , 16
V
 1 , 0
""
 1 , 0
""
 1 , 0
"TYPE"
 0 , 2
V
 0 , 2
V
 0 , 3
V
 0 , 3
```

```
V
 0 , 2
V
 0 , 2
V
 0 , 2
V
 1 , 0
""
 1 , 0
""
 1 , 0
"ENTRY_COUNT"
 0 , 440
V
 0 , 512
V
 0 , 512
V
 0 , 512
V
 0 , 512
V
 0 , 510
V
 0 , 510
V
 0 , 437
V
 0 , 73
V
 0 , 73
V
 0 , 73
V
 0 , 73
V
 0 , 72
V
 0 , 72
V
 0 , 5
V
 0 , 5
V
 1 , 0
""
 1 , 0
""
 1 , 0
```

```
"UTIL."
 0 , 0.921
V
 1 , 0
""
 1 , 0
""
 1 , 0
"CONT."
 0 , 0
V
 0 , 0
V
 1 , 0
""
 1 , 0
""
 1 , 0
"REMAIN."
 0 , 0
V
-1 , 0
BOT
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
"B:REPORT.GPS."
 1 , 0
""
 1 , 0
"FACILITIES"
 0 , 1
V
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
```

73

```
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
"CURRENT_COUNT"
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 2
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 1
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  1 , 0
""
  1 , 0
```

```
""
 1 , 0
"AVE._TIME"
 0 , 302.86
V
 1 , 0
""
 1 , 0
""
 1 , 0
"ENTRIES"
 0 , 512
V
 0 , 73
V
 1 , 0
""
 1 , 0
""
 1 , 0
"MIN."
 0 , 0
V
-1 , 0
BOT
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
"(V"
 1 , 0
""
 1 , 0
"STORAGES"
 0 , 1
V
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
```

```
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
"RETRY"
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  0 , 0
V
  1 , 0
```

```
""
 1 , Ø
""
 1 , Ø
"AVAILABLE"
 Ø , 1
V
 1 , Ø
""
 1 , Ø
""
 1 , Ø
"ENTRIES(Ø)"
 Ø , 108
V
 Ø , 3
V
 1 , Ø
""
 1 , Ø
""
 1 , Ø
"MAX."
 Ø , 2
V
-1 , Ø
BOT
 1 , Ø
""
 1 , Ø
""
 1 , Ø
""
 1 , Ø
""
 Ø , 1.1
V
 1 , Ø
""
 1 , Ø
"FREE_MEMORY"
 Ø , 78400
V
 1 , Ø
""
 1 , Ø
""
 1 , Ø
""
 1 , Ø
```

```
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
" "
 1 , 0
```

```
""
 1 , 0
""
 1 , 0
""
 1 , 0
"OWNER"
 0 , 440
V
 1 , 0
""
 1 , 0
""
 1 , 0
"AVE.CONT."
 0 , 1.23
V
 0 , 3.46
V
 1 , 0
""
 1 , 0
""
 1 , 0
"ENTRIES"
 0 , 512
V
-1 , 0
BOT
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
"#"
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
```

```
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
```

```
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
"PEND"
 0 , 0
V
 1 , 0
""
 1 , 0
""
 1 , 0
"AVE.TIME"
 0 , 57.47
V
 0 , 1136.12
V
 1 , 0
""
 1 , 0
""
 1 , 0
"AVL."
 0 , 1
V
-1 , 0
BOT
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 0 , 30712
V
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
```

```
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
 1 , 0
```

```
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
"INTER"
 0 , 0
V
 1 , 0
""
 1 , 0
""
 1 , 0
"AVE.(-0)"
 0 , 72.83
V
 0 , 1184.81
V
 1 , 0
""
 1 , 0
""
 1 , 0
"AVE.C."
 0 , 1.92
V
-1 , 0
BOT
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 0 , 2
V
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
```

```
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
" "
  1 , 0
  1 , 0
```

```
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
"RETRY"
 0 , 0
V
 1 , 0
""
 1 , 0
""
 1 , 0
"RETRY"
 0 , 0
V
 0 , 0
V
 1 , 0
""
 1 , 0
""
 1 , 0
"UTIL."
 0 , 0.96
V
-1 , 0
BOT
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 0 , 14
V
 1 , 0
""
 1 , 0
""
 1 , 0
```

""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0
""
  1 , 0

```
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
"DELAY"
 0 , 0
V
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
""
 1 , 0
"RETRY"
 0 , 0
V
-1 , 0
EOD
```

Enhanced simulation output through

the use of spreadsheet software

by

John Patrick Dieltz

Major, USAF

1986

86 pages

MASTER OF SCIENCE

Iowa State University
Ames, Iowa

BIBLIOGRAPHY

1. Banks, J. and Carson, J. S., II. "Process-interaction
    Simulation Languages." Simulation 44 (May, 1985):
    225-235.

2. Carroll, T. O. and Carroll, D. W. "Network Analysis
    with Spreadsheets." Lotus 1 (December, 1985): 70-74.

3. CINEMA. State College, PA: Systems Modeling
    Corporation, 1985.

4. DIF Technical Specification. Newton Lower Falls, MA:
    Software Arts Products Corporation, 1983.

5. Favreau, R. R.; Murtha, S. A. and Marr, G. R., Jr.
    "Desk-top Simulation - A State-of-the-Art Perspective."
    The Proceedings of the 1984 Summer Computer Simulation
    Conference, pp. 217-224. La Jolla, CA: Society for
    Computer Simulation, 1984.

6. Feltner, C. E. and Weiner, S. A. "Models, Myths and
    Mysteries in Manufacturing." Industrial Engineering 17
    (July, 1985): 66-76.

7. Friedman, R. "Ideas and Trends." Lotus 1 (December,
    1985): 24.

8. GPSS/PC User Manual. Stowe, MA: Minuteman Software,
    1985.

9. Haynes, J. L. "Circuit Design with Lotus 1-2-3."
    Byte 10 (Special IBM Issue, 1985): 143-156.

10. Kachitvichyanukul, V. and Lloyd, C. M. C. "A
    Microcomputer-based Integrated Simulation Support
    (MISS) System." The Proceedings of the 1985 Summer
    Computer Simulation Conference, pp. 9-13. San Diego:
    Society for Computer Simulation, 1985.

11. Kaplan, S. J. "Some Future Trends in Spreadsheets."
    Lotus 2 (February, 1986): 138.

12. Karian, Z. A. "GPSS/PC." Byte 10 (October, 1985):
    295-301.

13. Lashway, S. and Pendergast, N. "The New 1-2-3." Lotus
    1 (September, 1985): 60-64.

14. LeBlond, G. T. and Cobb, D. F. Using 1-2-3.
     Indianapolis: Que Corporation, 1983.

15. Lotus 1-2-3 User's Manual. Cambridge, MA: Lotus
     Development Corporation, 1983.

16. Macintosh, J. B.; Hawkins, R. W. and Shepard, C. J.
     "Simulation on Microcomputers: The Development of a
     Visual Interactive Modeling Philosophy." Proceedings
     of the 1984 Winter Simulation Conference, pp. 531-536.
     Piscataway, NJ: Institute of Electrical and Electronic
     Engineers, Inc., 1984.

17. Masri, S. and Moodie, C. "Using an Electronic
     Spreadsheet to Analyze Manufacturing Flow Systems."
     Computers and Industrial Engineering 9 (2) (1985):
     183-193.

18. McMillan, D. A. "Publisher's Letter." Lotus 1
     (December, 1985): 4.

19. Minor, R. J.; Sabuda, J. D. and Wortman, D. B.
     "Graphics and Simulation: Tools and Applications."
     Proceedings of the 1982 Winter Simulation Conference,
     pp. 161-171. Piscataway, NJ: Institute of Electrical
     and Electronic Engineers, Inc., 1982.

20. Morse, L. and Whitehouse, G. E. "Electronic
     Spreadsheets Evaluated for IE Applications."
     Industrial Engineering 17 (February, 1985): 17-26.

21. Morse, L. and Whitehouse, G. E. "IEs Must Look at
     Equipment, Needs in Choosing Spreadsheets." Industrial
     Engineering 17 (March, 1985): 21-31.

22. Mourant, R. R. "Turbosim - A Simulation Language for
     Microcomputers." The Proceedings of the 1985 Summer
     Computer Simulation Conference, pp. 26-31. San Diego:
     Society for Computer Simulation, 1985.

23. Parrinello, J. "Using the Personal Computer to Project
     Space Requirements." Computers and Industrial
     Engineering 9(4) (1985): 347-355.

24. Pritsker, A. A. B. Introduction to Simulation and SLAM
     II. Second edition. New York: Halsted Press, 1984.

25. Reese, R. M. "Simulation Programming Using Spreadsheet
     Software." Proceedings of the 1984 Winter Simulation
     Conference, pp. 539-542. Piscataway, NJ: Institute
     of Electrical and Electronic Engineers, Inc., 1984.

26. Schriber, T. J. *Simulation Using GPSS*. New York: John Wiley and Sons, 1974.

27. Standridge, C. R.; Hoffman, J. R. and Walker, S. A. "Presenting Simulation Results With TESS Graphics." *Proceedings of the 1984 Winter Simulation Conference*, pp. 305-311. Piscataway, NJ: Institute of Electrical and Electronic Engineers, Inc., 1984.

28. Vasudev, V. K. and Pruett, J. M. "Simulation on Three Sizes of Computers - An Evaluation." *Computers and Industrial Engineering* 9(3) (1985): 263-271.

29. Yancey, D. P. and Phillips, D. T. "Simulation Output Analysis within IDSS Prototype (2.0)." *1983 Annual Industrial Engineering Conference Proceedings*, pp. 471-479. Norcross, GA: Institute of Industrial Engineers, 1983.

END

DTIC

9 — 86