INTRODUCTION

Many papers oriented to the computer user deal with programming languages. These languages may be either flexible or oriented toward a particular problem field, such as military information retrieval or simulation; however, they are languages requiring the user to learn vocabulary, grammar, punctuation, and spelling to translate his problem into the specific language. This is not easy and generally requires considerable practice.

An officer on the communications staff of a military headquarters does not have time to study a language and learn how to express himself in it. In addition, he does not have the experience of an industrial engineer who is accustomed to flow-charting the operations needed to accomplish a function. Nevertheless, he needs to evaluate the effectiveness of his present methods and procedures and level of staffing under conditions which would occur when the workload might suddenly change because of world or local military or political events. He also needs to be able to determine whether any changes in methods, procedures, or staffing will improve the total response of the system.

The prime criterion for evaluation of a communication system is message transit time. Within this criterion are subcriteria to be chosen by the headquarters involved, which may specify:

The maximum transit time for messages of a specific class shall be less than $T$ minutes.

The percentage of messages of a specific class with transit time less than $T$ minutes shall be greater than $P$ percent.

Transit time through a system depends on two factors: processing time and waiting time. Processing time can be determined without the use of computers by observing the required time to perform specific tasks and by summing this time over all the tasks to be performed on a specific message. Waiting time is either batching or queuing time. Batching time is the time an operator waits after completing one task on a message before delivering it to the next task or operator, so that the first operator can continue performing the same task on a number of messages; this time can be estimated. Queuing time can be mathematically estimated when only a few queuing points are involved. However, when many dynamically interacting queues must be considered, Monte Carlo simulation techniques must be used to gather information about the formation of queues and the delays caused by queuing. This requires digital-computer simulation.

The Franklin Institute Research Laboratories (FIRL) has developed two tools for officers on the communication staff of a military headquarters to use for system evaluation; these tools were developed as part of a study for the Department of the Army and the Defense Communications Agency to improve message processing operations within a headquarters. One tool is a method called Auto-
matic Flow Process Analysis (AFPA) which allows personnel without any flow-charting or system-analysis experience to develop accurate flow charts by carrying out a set of procedures. The second is Simplified Message Processing Simulation (SMPS), with which the same personnel can prepare a simulation model and message samples by following a set of simply stated procedures; SMPS does not require personnel to learn any programming language.

With SMPS, members of a military communications staff can evaluate a message-processing system under dynamic conditions without requiring the services of personnel experienced in computer technology or programming. The SMPS toolbox contains building blocks and a framework with which a model of a message-processing system can be built.

COMMUNICATIONS STAFF NEEDS

The communications staff at a military headquarters needs to be continually aware of the capabilities and effectiveness of their current message-processing systems, not only with respect to current traffic but also with respect to crisis conditions which may occur. Figure 1 shows an overview of the activities within a Message Communications Terminal office (communications center and staff message control) at a military headquarters. Within the limits of military regulations and command structure, this staff is able to suggest changes to improve system operation. However, changes should not be implemented unless there is assurance that the total system operation will be improved; therefore, methods for evaluation are required. Because of the differences in needs, regulations, and traffic at different headquarters, only the staff at the individual headquarters (rather than a higher agency) can best evaluate its own systems. The likelihood that these operational staff personnel have programming background or inclination is very small.

TOOL 1, AFPA

AFPA permits the non-system analyst to construct an accurate flow chart of the operation of his system. In the message-processing case for which AFPA was designed, the message passes along the flow of the chart through the tasks performed in the boxes of the flow chart. A task is called an event and specifies what personnel and equipment are involved (such as a communications-center receive operator or a Xerox machine), what is done (such as tearing the message from the teletype monitor), and how long the event takes (such as 30 seconds).

TOOL 2, SMPS

Figure 2 shows a small fragment of an AFPA flow chart. A detailed simulation including all of
the events on an entire AFPA flow chart, would take many hours to construct; however, Simplified Message-Processing Simulation permits the AFPA events to be grouped into broader tasks which may be matched directly to the SMPS building blocks. Thus, a simulation model may be assembled quickly and easily. The outlined area of Fig. 3 is the SMPS simplification of the outlined portion of the AFPA flow chart fragment shown in Fig. 2.

A technical report, "SMPS—Simplified Message-Processing Simulation," instructs the user how to construct simplified flow charts from the AFPA flow charts, how to fill out task-description worksheets from the simplified flow chart, and how to match the SMPS building blocks to the tasks defined.

When the SMPS building blocks are matched to the tasks, the simulation model is essentially complete. The SMPS report also describes how the input messages for the simulation may be prepared from a real traffic sample or from statistically generated messages.

**What SMPS Is**

SMPS is a language derived from the macro-assembler capabilities of IBM's GPSS II. The building blocks of SMPS include a set of GPSS variables defined in terms of the parameters of a GPSS transaction, a set of functions for the generation of GPSS parameters from a deck of cards generated independently to describe a message sample, a few other GPSS system variables, and a set of GPSS macro instructions. SMPS relies heavily on the development of DMPS (Detailed Message-Processing Simulation) for the method of parameter construction and assignment.
Use of SMPS

The simplified flow chart is derived from the AFPA flow chart of chains of events, which is in tree form. First, uninterrupted tasks are identified, and identical chains of tasks are merged. The first worksheet describes personnel and equipment (Fig. 4). The second worksheet describes the tasks in the simplified flow charts in terms of personnel and equipment required, next tasks to be performed under what conditions, and processing time expected, either in numbers or as a formula in terms of message characteristics (Fig. 5). The next task is the selection of SMPS building blocks. In the simplest cases, a SMPS building block matches each task. However, if the task is complex or unusual, it may be necessary to divide it into several simpler tasks to find a match.

In the case of communications-terminal processing for which SMPS was designed, the usual tasks are decision-making, logging, routing, poking (tape cutting), tape reproduction, offline encryption/decryption, inspection, transmission, filing, reference lookup, transportation or delivery between major staff areas of a headquarters, typing, reproduction, collation, distribution of copies, and additional administrative functions.

The SMPS building blocks, called modules, are divided into six categories. The first category contains general-purpose modules which involve queue number, personnel or equipment identifica-
Figure 4. Worksheet 1, personnel and equipment.

tion, next module, and time factors; the time factors are for processing or batching. The second category contains decision modules for usual decisions. The third category contains modules which represent transportation and contain facilities to record transit times within the GPSS simulation and for external statistical analysis. The fourth category contains a set of modules to permit delivery at regular intervals. The fifth group contains the modules used to control the flow of the three types of messages into the model. The first type of message is the sample concerning which transit times are to be measured to evaluate system effectiveness; this is the group which is specified on a card deck generated independently of the model. The second type concerns service and similar messages which are not being directly evaluated but which occupy both personnel and equipment within the communications facility; these are an integral part of the model. The third type of message does not represent actual traffic but is used to account for any other activities, such as breaks for personnel or downtime for equipment, which would impede the processing of significant message traffic by occupying personnel or equipment and making them unavailable. The sixth group contains flexible modules which allow most unusual tasks to be performed without requiring knowledge of GPSS.

Figure 6 contains the description of a few modules.
<table>
<thead>
<tr>
<th>Task ID Code</th>
<th>Task Description</th>
<th>Personnel and Equipment Needed</th>
<th>Processing Time</th>
<th>Next Task</th>
<th>Queue Discipline</th>
<th>Output Batched</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Tear off message from TT receiver</td>
<td>SMC Inlexer</td>
<td>AFPA Event</td>
<td>Sec.</td>
<td>J</td>
<td>FIFO</td>
</tr>
<tr>
<td>J</td>
<td>Decision flash or not</td>
<td>SMC Inlexer</td>
<td>005</td>
<td>33</td>
<td>J</td>
<td>FIFO</td>
</tr>
<tr>
<td>G</td>
<td>Set queue discipline to FIFO</td>
<td>SMC Inlexer</td>
<td>020</td>
<td>1</td>
<td>0</td>
<td>FIFO</td>
</tr>
<tr>
<td>B</td>
<td>Administrative processing flash message</td>
<td>SMC Inlexer</td>
<td>035...105</td>
<td>41</td>
<td>A</td>
<td>FIFO</td>
</tr>
<tr>
<td>A</td>
<td>Routing of Messages SMC</td>
<td>Message Controller</td>
<td>140...185</td>
<td>309</td>
<td>Off chart</td>
<td>FIFO</td>
</tr>
<tr>
<td>H</td>
<td>Set queue discipline to FIFO</td>
<td>SMC Inlexer</td>
<td>010</td>
<td>1</td>
<td>0</td>
<td>FIFO</td>
</tr>
<tr>
<td>F</td>
<td>Trim and splice message</td>
<td>SMC Inlexer</td>
<td>110...015</td>
<td>30</td>
<td>E</td>
<td>FIFO</td>
</tr>
<tr>
<td>E</td>
<td>Decision immediate or not</td>
<td>SMC Inlexer</td>
<td>020</td>
<td>1</td>
<td>0</td>
<td>FIFO</td>
</tr>
<tr>
<td>D</td>
<td>Administrative processing immediate message</td>
<td>SMC Inlexer</td>
<td>120...105</td>
<td>89</td>
<td>A</td>
<td>FIFO</td>
</tr>
<tr>
<td>C</td>
<td>Administrative processing low precedence message</td>
<td>SMC Inlexer</td>
<td>065...105</td>
<td>24</td>
<td>A</td>
<td>FIFO</td>
</tr>
</tbody>
</table>

**Figure 5. Worksheet 2, task definition.**

<table>
<thead>
<tr>
<th>Name</th>
<th>No. of Variables</th>
<th>Module Description</th>
<th>Meaning of Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>5</td>
<td>General</td>
<td>Queue No.</td>
</tr>
<tr>
<td>B1</td>
<td>7</td>
<td>General</td>
<td>Queue No.</td>
</tr>
<tr>
<td>B2</td>
<td>8</td>
<td>General</td>
<td>Queue No.</td>
</tr>
</tbody>
</table>

**Figure 6. Selection of available modules.**
The cards representing the significant sample are prepared by a computer program to a form acceptable by the simulation program; these cards contain the identification and significant characteristics of each message. The computer program (written in FORTRAN) also prints these characteristics of each message in English (Fig. 9). This printout includes time of arrival in the system as day, hour, and minute, as well as the simulator clock time for arrival in total seconds. It also includes the identification number, which indicates whether the message is incoming or outgoing (those numbered over 20,000 are incoming), precedence, classification, number of addresses, number of lines of text, number of communications channels required, number of pages, number of staff agencies on local distribution, number of local copies, whether off-line encryption or decryption is required, special security categories, or other special characteristics involved.

Two programs are available to prepare a message deck.

One program uses an actual message sample, in which case the message characteristics are determined by examining a message. These characteristics then are transcribed onto cards, which are used as data by this input-preparation program. If statistical generation of messages is desired, an

![Worksheet 3, module assignment.](image)

Worksheet 3 (Fig. 7) aids in the matching of modules to tasks. The function-definition worksheet (Fig. 8) aids in the construction of functions to define processing times in terms of message characteristics.

![Function-definition worksheet.](image)
alternative FORTRAN input-preparation program is available with which the message characteristics necessary for the run can be easily specified. A listing of these specifications and detailed diagnostic routines concerning card or logical errors is provided. The other outputs of this program are the same as those of the first input program.

Relatively few items in the printout of the simulation run are significant to this type of model. Hence, the volume of printout to examine is not excessive.

The most important question in evaluating a model of a message-processing system is, “How long does it take a message to get through the system?” This information is most meaningful in terms of the cumulative distribution function of the total transit time through the system; however, it may also be important to know the time through major subsystems, as well as the time for messages with special characteristics.

A major output of SMPS is a deck of cards, each of which contains all the characteristics of a message, a transit time either through the entire system or through a major portion of the system, and an identifier specifying the meaning of the transit time given. Thus, this card deck can be processed manually, by EAM equipment, or by computer to select the messages with the characteristics of interest and to determine the transit-time distributions for these characteristics.

The other output of SMPS is the printout produced by the GPSS program. The portions of the output significant to the user include the tables which give the fraction of total number of messages with transit times less than each increment of an accumulating time scale, and the queue statistics which indicate where bottlenecks occur.

**RANGE OF APPLICABILITY OF THE PRESENT PACKAGE**

Although this application is based on AFPA flow charts, the technique does not require that AFPA be used. The flow chart which describes system operation may be constructed independently; however, in this case, more skill may be required in defining the tasks of suitable size. The basic concept is that a task must be small enough that the personnel and equipment involved would not be interrupted to perform any service for any other message.

The basic structure of SMPS assumes that a message has certain properties which are recorded in the simulation representation of the message—namely,
the GPSS transaction parameters. Two properties
are not fixed and may be defined at each head­
quartes; however, these properties may have, at
most, 10 values. The characteristics chosen are
ones most meaningful to a variety of military head­
quartes. Hence, although the processing examples

carried out thus far involved military terminal proc­
cessing, SMPS should be useful for any processing of
military or nonmilitary messages. Although such
properties as off-line encryption or security classi­
fication are not apt to be meaningful for nonmili­
tary applications, any properties defined can be ig­
nored in a model. If the statistical input program
is used, each specified characteristic must be ex­
amined to determine whether it can be ignored in
creating the message sample.

In its current form, SMPS can be used to simulate
any message-processing application where the
transit time for a message and its flow through the
processing steps depend only on the message char­
acteristics defined in SMPS and on statistical
variables.

APPLICABILITY OF TECHNIQUE
FOR OTHER USES

The SMPS technique is not limited to dynamic
analysis of message processing. Whenever a system
can be looked on as consisting of processing units
which can be described by a small number of char­
acteristics and where both processing time and
batching time depend on characteristics of these
units alone, a set of building blocks and a structure
similar to SMPS can easily be constructed in a
very short time by personnel with programming
experience.

Because most flow charts contain relatively few
patterns of boxes and lines, it is possible to de­
scribe most systems by reusing a few modules with
different variable values. For example, one general
equipment- or personnel-use module can be used
which includes as variables a queue number, three
or four equipment/personnel identities, the next
task, and several time factors. Two decision
modules corresponding to two- and three-path
branchings, will probably be sufficient. Decision
modules have variables of relations (less than, equal
to, greater than, for example), a number being
tested by the relation, next task if true, and next
task if false. A few special modules can be pro­
gammed to insert in the flow-process chart for
priority assignments, tabulations, origination rates,
and the like. With these types of modules, a model
can easily be constructed.

ADVANTAGES OF THE SIMPLIFIED
MESSAGE-PROCESSING SIMULATION

A “language” such as SMPS is easier to learn
than a simulation or programming language: it
has no grammar and little vocabulary. A model in
SMPS can be constructed very quickly. Changes
are readily made and alternatives are easily com­
pared. Because the level of abstraction is high, the
model is easily understood in terms of activities
which occur and of what is required for the activ­
ities.

REFERENCES

ing (IMP)—An Analysis of Headquarters Message­
Processing Operations,” Technical Report 1-055,
Franklin Institute Research Laboratories (Oct.
1965).
2. P. W. Maraist and A. Barskis, “Automated
Flow Process Analysis (AFPA)—A Technique for
Analysis of Headquarters Message Processing,”
ibid, no. 1-160 (Nov. 1965).
Message-Processing Simulation (SMPS)—A Tech­
nique for Analysis of Headquarters Message Proc­
cessing,” ibid, no. 1-161 (Nov. 1965).
4. General Purpose Systems Simulator II, Form
B20-6346-1, International Business Machines Cor­
poration (1963).