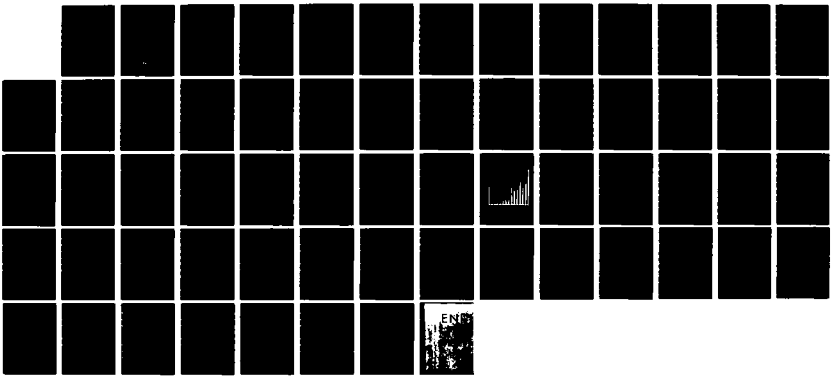
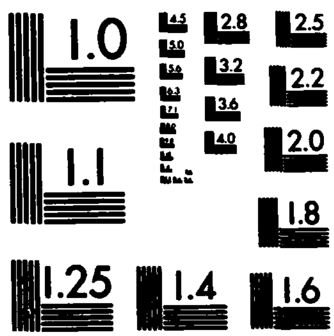
AD-A133 518 A PERFORMANCE-BASED TECHNIQUE FOR ASSESSING EQUIPMENT MAINTAINABILITY(U) UNIVERSITY OF SOUTHERN CALIFORNIA REDONDO BEACH BEHAVIORAL TE.. D M TOWNE ET AL. AUG 83 1/1

UNCLASSIFIED TR-102 N00014-80-C-0493 F/G 15/5 NL





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

A PERFORMANCE-BASED TECHNIQUE FOR ASSESSING EQUIPMENT MAINTAINABILITY

Douglas M. Towne
Mark C. Johnson
William H. Corwin

August 1983

Technical Report No. 102

BEHAVIORAL TECHNOLOGY LABORATORIES

Department of Psychology

University of Southern California

Sponsored by

The Engineering Psychology Group
Office of Naval Research

Under Contract No. N00014-80-C-0493
ONR NR196-165

A PERFORMANCE-BASED TECHNIQUE FOR ASSESSING EQUIPMENT MAINTAINABILITY

Douglas M. Towne
Mark C. Johnson
William H. Corwin

August 1983

Technical Report No. 102

BEHAVIORAL TECHNOLOGY LABORATORIES

Department of Psychology

University of Southern California

Sponsored by

The Engineering Psychology Group
Office of Naval Research

Under Contract No. N00014-80-C-0493
ONR NR196-165

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	
2. GOVT ACCESSION NO.	AD-A133518
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	A PERFORMANCE-BASED TECHNIQUE FOR ASSESSING EQUIPMENT MAINTAINABILITY
5. TYPE OF REPORT & PERIOD COVERED	Final Report
6. PERFORMING ORG. REPORT NUMBER	Technical Report No. 102
7. AUTHOR(s)	Douglas M. Towne, Mark C. Johnson, and William H. Corwin
8. CONTRACT OR GRANT NUMBER(s)	N00014-80-C-0493
9. PERFORMING ORGANIZATION NAME AND ADDRESS	Behavioral Technology Laboratories University of Southern California 1845 S. Elena Ave., Redondo Beach, CA 90277
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	NR 196-165
11. CONTROLLING OFFICE NAME AND ADDRESS	Office of Naval Research Engineering Psychology Group (Code 442EP) Arlington, VA 22217
12. REPORT DATE	August 1983
13. NUMBER OF PAGES	54 + -iv-
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	
15. SECURITY CLASS. (of this report)	Unclassified
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)

18. SUPPLEMENTARY NOTES

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

Maintainability
Maintenance Performance
Troubleshooting
Fault Isolation
Diagnosis

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

Please see reverse side.

Previous research produced a computer-implemented model of corrective maintenance performance, based on a relatively simple maximum-productivity rule for selecting maintenance operations, and a relatively complex data base representing particular systems.

The model has been expanded to recognize the impact of task-sequence context upon the actions necessary to accomplish tasks. Decisions at each stage of a simulated maintenance requirement now reflect the effects of previously performed work on the time and effort necessary to perform future tasks.

Maintainability projections were generated for a digital infrared transmitter/receiver system, specially constructed to be configured in two functionally equivalent forms. Ten electronics technicians worked to identify and resolve eight inserted malfunctions each, using built-in indicators and standard test equipment. The overall projections of maintenance times compared well to the experimental data.

A measure of design complexity is proposed for the evaluation of maintainability. This measure, mean number of indicators necessary to accomplish fault isolation, is sensitive to multiplicity of fault modes and to the extent to which fault symptoms are confounded at the maintainer interface.

ACKNOWLEDGEMENTS

This research was sponsored by the Engineering Psychology Group, Office of Naval Research, Mr. Gerald S. Malecki serving as scientific officer. We wish to thank them for their support of this work.

We also thank Lt. Cmdr. Charles Walker, Officer in Charge, Advanced Electronics School Department, and ETCM Raymond for their cooperation and support of the maintenance experimentation conducted at their facility. The experimentation was conducted by Mr. Richard Mishler of our staff.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION I. INTRODUCTION

Background

The objective of this research is to develop a technique for assessing the inherent maintainability of an equipment or system based upon specifications of the system design. The development and application of such a technique would offer a quantitative basis upon which to judge the relative merits of alternative designs. Such a tool could be applied during the design cycle to evaluate the benefits of various design options, and it could be applied at the procurement stage to compare the expected maintenance workloads imposed by competing design concepts. In a longer-range role, such a resource could be employed as a human factors research aid, to explore the impact of design on maintainability over a range of design variables.

Scope

As maintainability is a measure of human performance in relation to a hardware design, the assessment technique is performance-based, i.e. it has the goals of projecting what actions would be performed to meet specific corrective maintenance requirements, and of quantifying the times to perform the projected action sequences. Preventive maintenance can be considered a sub-set of corrective maintenance, for our purposes, since the operation sequences may be determined without engaging a model of fault-isolation behavior. The technique is not a model of human decision-making processes. Rather, it is a computer-implemented process for selecting operations to isolate and resolve faults, based upon a maximum-productivity criterion. The data base upon which this process operates, however, was devised to reflect the kinds of equipment-specific information available to a human technician.

As described below, certain steps were taken to expand the model's initial objective function, so that it considered real-world factors such as hardware cost and the urgency of the maintenance situation. As a result, the testing sequences generated by the process were brought more in line with those produced by human technicians operating in similar situations.

The ultimate objective is to project and quantify both manual and cognitive processes, as they are applied to equipment maintenance. For the present, we are concerned primarily with projecting representative sequences of manual actions, and quantifying the time to perform those sequences.

Previous Research

During development, consistent differences between the model's testing decisions and those made by real technicians were identified and rectified. Briefly, those differences resulted primarily from two major sources: First, the original model contained no consideration of hardware cost, and thus it replaced a unit whenever the replacement, followed by a confirming test, provided information faster than would the tests necessary to check the suspected unit. In other words, the model behaved as one might in an urgent situation where equipment restoration takes precedence over all other issues, including the cost of replacement units.

This was rectified by adding hardware cost to the data base, and by introducing two parameters which characterize the maintenance environment. The first parameter expresses the urgency of the maintenance situation; the second expresses the extent to which hardware is available for substitution. In urgent environments, or in situations where replacement parts are inexpensive, the model will use replacement as an effective means of checking a unit, rather than expending the time to perform functional checks. Otherwise, it will continue with conventional information-gathering operations.

The second characteristic of the original model was that its memory (the computer's memory) was extensive, highly detailed, and flawless, resulting in three areas of dissimilarity with human technicians, as follows:

1. The detailed fault-effects data, representing the model's perfect knowledge of the system, were complete and accessible at all times, to guide test selection and symptom interpretation.

2. All possible faults were considered in determining the values of the tests under consideration. This led to uncharacteristic testing sequences which reflected little continuity of purpose.

3. The best possible test was always chosen, based upon a quantitative ranking of their relative values; a test whose calculated value ranked a close second was never selected.

These capabilities are desirable and impressive, but they are not characteristic of human abilities and approach. Two steps were taken to address these problems. First, the model's fault effects knowledge was made less detailed. This was accomplished by combining the many distinct symptom types into just four descriptive categories, as follows:

<u>Category</u>	<u>Failure effect(s)</u>
a	Failure of unit <i> will not affect indicator <j>. This is indicated in the fault-effects matrix by a '0' at row i, column j.
b	Failure of unit <i> will affect indicator <j>; the symptom will be symptom <k>, no matter how the unit fails. This is indicated in the fault-effects matrix by a <k> in row i, column j (k ranging from 1 to 6).
c	Failure of unit <i> may affect indicator <j>, depending upon the fault mode. This is indicated in the fault-effects matrix by a '7' at row i, column j.
d	Failure of unit <i> will affect indicator <j>; the symptoms vary, depending upon the fault mode. This is indicated in the fault-effects matrix by an '8' at row i, column j.

It is important to note that this process for representing fault effects in the model's data base is not arbitrary, and that the uncertainty, or fuzziness, exhibited by the representation scheme reflects the extent to which fault effects are confounded as a result of the system design. When failures in a system produce effects which map closely to the replaceable elements, then that system's data base will reflect clear and easily identifiable symptoms.

Ideally, a system would be designed so that all failure effects are of types 'a' or 'b', above, and the symptom patterns for all replaceable elements are unique. For less ideal systems, replaceable units will exhibit multiple failure modes, and the symptom patterns will be highly confounded. The maintenance performance projected by the model for such systems will reflect the increased difficulty of fault isolation.

The second remedy was to limit the model's 'interest' in suspected elements at each stage, to only those elements whose likelihood of causing the obtained symptom patterns was at least one-half the likelihood of the most suspected element. Tests were therefore evaluated based upon their information value relative to these few replaceable units. This modification had the effect of causing the model to focus on a few suspected elements, and to conduct testing to check out those units before initiating a new line of inquiry. This 'hypothesis testing' attitude conformed well to the observed troubleshooting performances.

Current Research

The developments described above ultimately produced a model which generated maintenance time distributions very similar to ones obtained experimentally (Towne, Johnson, and Corwin, 1982). The issue of validity was not tested by those results, however, as the model was refined to conform to existing data. The study described in this report was conducted to provide a first real test of the model in a realistic maintenance environment.

Organization of the Report

Section II of this report presents a summary of the performance model, including its scope, organization, test-selection algorithm, and process for generating conditional task sequences.

Section III presents data from a study of corrective maintenance performance, compared to projections of the performance model.

Section IV defines and discusses Maintenance Complexity, a system complexity measure related to ease of fault isolation.

Final conclusions and plans for future work are presented in the final section.

SECTION II. THE MAINTENANCE PERFORMANCE MODEL

The projection technique is embodied in a computer model of troubleshooting performance, termed PROFILE (Towne, Johnson, & Corwin, 1982). PROFILE generates testing sequences to isolate and resolve specified faults in a hardware system, the design of which is represented in a digital data base. The data base for any particular system itemizes that system's replaceable units, the available test points and built-in indicators, the possible effects of failures of the replaceable units, and the physical structure of the design as it affects accessibility to the test points and replaceable units.

Scope of Application

The central task of PROFILE is to generate a sequence of steps to isolate and rectify a fault in a system. Multiple sequences may be produced, for any fault, by running the program in a sampling mode, and many faults may be so analyzed. These sequences yield projected frequencies with which various maintenance actions would be performed. Some maintenance procedures will be prescribed by technical documentation, and perhaps constrained by automated sequences. In these cases the work content can be determined directly from the procedural instructions and then quantified similarly to PROFILE-generated sequences.

The manual times to perform each task sequence are computed by accumulating time values for each generated maintenance action. A maintenance action is considered to be a short work element which is performed in a relatively consistent manner, regardless of the context in which it occurs. For example, placing a probe on a test point and observing a meter would be considered an action since the variations possible as a consequence of sequence context are minor. Times for actions are relatively fixed, and can therefore be retrieved from a data bank of 'standard' times. These time values are pre-derived using classical industrial engineering work measurement techniques of time and motion study. A major task for the performance model is to determine what actions must be performed to accomplish higher-level goals, or operations.

The quality of the model's projections depend upon the appropriateness of its projected troubleshooting sequences, and upon the accuracy of assigning time values to the tasks in those sequences. Of these two issues, the work content of the generated sequences is by far the more critical. While different technicians will work at varying workpaces, these variations are quite small compared to the variability in the operation sequences they perform during self-directed fault isolation. Consequently, fixed values are considered adequate for quantifying the performance times of subtasks.

The Model's World-View

The general structure of the model is a hierarchy of rules, expressed in the Pascal programming language. The highest-level rule, which affects nearly all of the model's decisions, is that maintenance operations under consideration are preferred in relation to their expected productivities. The productivity of a test is computed as its expected fault isolation value divided by the time to perform it.

Undoubtedly, other concerns can dominate a technician's approach to a maintenance requirement. Avoidance of danger, discomfort, or catastrophic error are almost certain to play major roles in maintenance of military systems. Additionally, scarcity of spares and extreme time constraints may greatly distort the approach a technician would otherwise pursue. These environmental conditions could greatly decrease the attractiveness of certain maintenance operations, possibly to the extent that the actions are avoided altogether.

The criterion of productivity, or maximum rate of progress per unit time, can function in the face of these seemingly pathological factors. In the model's terms, extremely high (possibly infinite) time penalties would be associated with those maintenance actions which could seriously harm either the equipment or the maintainer. A future version of the current model might include the effects upon expected performance time of environmental factors and human error.

The performance model generates a sequence of operations to identify and rectify any hypothesized fault. Each selected operation offers the potential of providing new information about the system. If the operation involves checking one or more indicators, new information may be obtained from the symptoms. If the operation involves a replacement or adjustment, some fault possibility may be eliminated from consideration (the possibility of faulty replacement parts is not currently considered).

This general view of a maintenance operation allows the model to consider, in a uniform manner, the relative merits of performing conventional tests, replacements, and adjustments.

As described later, the model considers the preconditions which must be satisfied to perform any maintenance operation, and it recognizes the conditions established by prior operations. Examples of preconditions include the following:

- partial disassembly of a unit, to gain access to a test or adjustment point, or to replace or repair a system element;
- partial or total reassembly of a unit, to perform a test or adjustment;
- equipment reconfiguration, to perform a subsequent test under different conditions.

The model evaluates the preconditions which are required by each maintenance operation, and it generates action sequences to achieve the necessary conditions. Design characteristics which affect the ease of testing, reconfiguring, replacing, and adjusting all impact the maintenance sequence at each stage of the process. The model would therefore be sensitive to such design decisions as moving a test point, adding a fastener, or modularizing a group of components.

Following a replacement, the model performs the shortest available test which is sure to be abnormal if the fault persists. This is often chosen to be some test which previously yielded an abnormal result (although if considerable reassembly is required some other test could be selected as being more efficient). The model continues to generate maintenance operations until the assumed fault has been rectified, by a replacement or adjustment, and normal system operation has been confirmed.

Organization of the Model

The maintenance performance model is organized on two levels, data and program (see Figure 1).

Data. The data level consists of two types of information:

1) equipment-specific information, i.e. the equipment design specification, which includes the following:

- the tests available in a system's design,
- the indicators which are involved in those tests
- the adjustment points and replaceable elements
- the possible effects of faults, on the indicators
- the times to perform the tests, adjustments, and replacements
- the relative costs and reliabilities of the replaceable elements

2) working memory, which reflects the current status of a maintenance problem in progress. The primary contents of working memory are likelihood measures for each replaceable unit and current values of various equipment conditions which can change during maintenance work.

The likelihood measures are derived from numerical scores, maintained for each possible fault or mis-adjustment. The score for a replaceable element reflects the difference between the symptoms already received in a problem and those which the element might have produced, if it were the

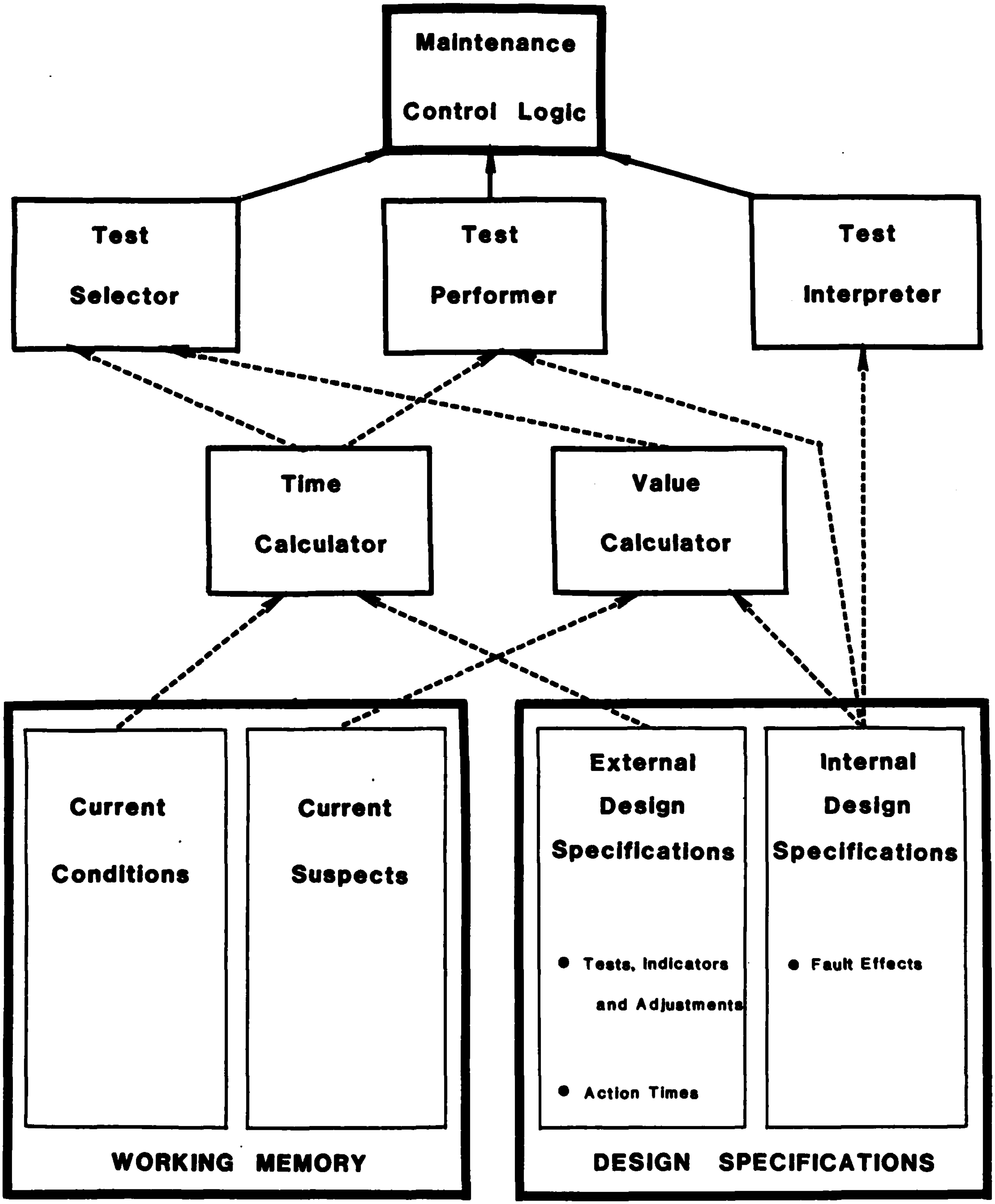


Figure 1. PROFILE System Organization

failure. Consequently, a relatively low score indicates a close fit, and a likely source of the symptoms received.

The equipment conditions reflect the status of various attributes of a system, such as the extent to which it has been disassembled. This information is maintained and accessed by the model to determine the workload involved in performing various possible maintenance operations.

<u>Program</u>. The computer program, written in Pascal, contains two levels of control mechanisms, although these are not clearly partitioned as separate entities. At the top level is what may be regarded as generic maintenance control and planning logic. This control structure invokes the lower-level functions in a relatively fixed sequence, as follows:

```
REPEAT (for each fault examined)

    1. SELECT THE NEXT OPERATION.
       (and add its context-dependent performance time
        to the cumulative total)

    2. If Adjustment or Replacement was selected in step 1:
       SELECT THE SHORTEST CONVENTIONAL TEST WHICH MONITORS THE SYSTEM.
       (and add its performance time to the total)

       else  DETERMINE THE OUTCOME OF CONVENTIONAL TEST.
       (by looking up the symptom produced by the true fault)

    3. UPDATE THE LIKELIHOODS OF THE POSSIBLE FAULTS.

UNTIL no fault
```

<u>Test Selector</u>

The general rules applied to select the next operation (step #1) consider all of the following:

- the relative reliabilities of the replaceable elements
- the costs of the replaceable elements
- the current likelihoods of the possible faults
- the times to perform the operations, in the present context
- the new information possibly obtained from the operations

The selected operation may be a replacement, as well as a test or adjustment, because replacements (followed by confirming tests) provide new information about the system. In early phases of problems, replacements are not usually attractive choices because they offer little information compared to other tests. In addition, their relative time investment is often large, since the time to obtain new information includes the time to access and replace the part plus the time to perform a confirming test. Later in problems, however, replacements may offer more information than the remaining available tests, for the time investment involved.

Like many human technicians, this decision logic may decide to replace a part prior to obtaining complete proof of its failure, simply because the time to replace is low compared to the time required to complete the necessary confirming tests. Excessive behavior of this type by human technicians is termed 'Easter-egging' and is costly unless the replacement parts are extremely inexpensive. The maintenance model considers component costs to avoid Easter-egging, but it will resort to swapping moderately-priced suspected elements if the times of the other useful tests are excessive, and the maintenance environment is sufficiently urgent.

The test selection algorithm forms an ordered set of n productive tests, with test 1 being the test of highest value, and n being a parameter set to less than, or equal to, the total number of tests available. Since only productive tests are included in the set, previously performed tests, and tests which have no new information to offer, are not considered for selection. The value of a test is computed as the ratio of the test's likely information value to its performance time. Both of these quantities are sequence-dependent and are re-computed at each stage. The values of the n possible tests are then normalized, and a test is selected probabilistically according to the relative values.

With n set to 1, the model selects the 'best' available test at each decision point. With larger n, poorer tests are considered for selection. Since probability of selection is related to computed value, however, extremely poor tests are rarely selected by the model. In previous studies, best results were obtained with n set equal to three.

Test Performer

The test performance function simply appends the selected operation to the ongoing sequence, adds its performance time to the cumulative total, and retrieves from the data base the symptoms which would be obtained if that test were actually performed.

Test Interpreter

The test interpretation function compares the symptom received from the test to the possible fault effects of the replaceable units. It computes a difference score for each unit, which varies according to the difference between the unit's possible fault effects and those received from the test performed. These values are then added to the cumulative distance scores, and normalized likelihoods are computed from the scores.

If the system's replaceable units can fail in multiple modes, the model may, at times, perform tests which turn out to yield no useful information. Suppose, for example, three units (A, B, and C) are suspected, based upon prior symptoms. Suppose further that only unit A could affect test 1, if it fails in a particular mode. If test 1 is performed, and an abnormal result is obtained, then the fault is known to be in unit A. If a normal result is obtained, however, nothing is learned except that part of unit A is operational. It is seen that normal results are useful for eliminating a unit from suspicion only if the element always produces an abnormal result on the indicator, when it fails, regardless of its failure mode. When replaceable units are functionally large, they tend to exhibit more failure modes, often including a mode in which there is no effect on indicators intended to monitor those units.

Consequently, the model finds normal symptom information to be less useful than abnormal symptoms, in many cases. Thus, if a designer were to revise the modularity and/or packaging of the replaceable units, the model would sense the fault isolation implications.

Generating Conditional Task Sequences

One of the prime factors considered by PROFILE in selecting maintenance operations is the time to perform the available alternatives. An 'operation', as used here, denotes the completion of some work, such as replacing a part, making a test reading, or adjusting a device. The particular actions performed to accomplish an operation can vary considerably, depending upon the conditions of the equipment existing as the operation is initiated. Consequently, operation times can vary significantly even when performed at a fixed workpace.

An operation may be considered as having two components, a fixed portion which is independent of context (and is always performed), and a conditional portion which is affected by previous work. The fixed component of an automated test, for example, might consist of the actions to key in a test number, wait for the test to be completed, and observe the outcome. The conditional portion might involve reconfiguring the operational system or the automated test unit, if necessary.

The effects of these conditional time variations are felt at many levels. When time dependencies are large, maintainers may consciously take advantage of an existing equipment state by obtaining as much information as possible before dismantling that state. Or, when normally inaccessible parts become exposed by previous actions, they may perform checks or replacements which would not have been otherwise undertaken. At the motion level, the effect may be to encourage the continued performance of a type of operation, such as making readings with an oscilloscope, once the setup has been accomplished for the first such test.

The recognition, in the model, of conditionalities thus promotes a type of 'inertia', which both encourages the continuation of a type of operation, and which promotes opportunistic lines of approach. Interestingly, these effects are realized as a result of the simple optimization criterion under which the model always operates, rather than being produced by specific rules or psychological models.

General Formulation

The model's data base has been expanded to accept conditional information which can be as extensive as necessary, or may be set to reflect a total lack of dependencies. To select the next maintenance operation, the model computes the time required to perform each one, starting from the current status of the system.

When even modest dependencies exist, the computational load of exploring all possible action sequences to perform an operation can become unmanageable. Consequently, the model accepts the first action sequence it can generate which will meet the operation's pre-conditions. In an attempt to generate rational and reasonably efficient sequences, the model applies a heuristic scoring procedure to the alternative actions at each stage, and selects the action which appears to be most appropriate. The search process can lead to 'dead-ends' in which an action sequence being generated cannot achieve the desired condition. In that case, the previous decision is revised, and the search for a workable sequence continues. The decision tree expansion and back-tracking is continued until an action sequence is generated which achieves the necessary pre-conditions for the operation at hand.

To compute conditional operation times, the model maintains a state vector, S1, S2 , ... ,Si , ..., Sn wherein each Si expresses the state of attribute i. An attribute is a particular aspect of a system which can change, and which affects operation times. An example attribute for automobile maintenance would be the status of the engine, and its two states would be RUNNING and NOT RUNNING (capitals will denote attribute states). Maintenance problems might begin in some standard state, say with the engine NOT RUNNING. Following the performance of any action which left the engine RUNNING, the model would mark that attribute as being in the RUNNING state. Any test which requires the engine to be NOT RUNNING then would require additional time to complete.

Each major maintenance operation may be assigned a state vector in the data base, where each Si expresses the state of attribute i which must be achieved in order to perform the operation. For example, a compression check or oil pressure check would require that the engine attribute be in the RUNNING state, whereas other actions might require that the engine be NOT RUNNING. Some actions might not be affected by the state of the engine, and would be so indicated.

The physical structure of a system design is reflected in these state vectors. Access to test points, modules, or individual components is recognized by the model, and it generates the appropriate disassembly or reassembly actions when required.

Each attribute may have a 'change-requirement' vector, similar to the performance-requirement vectors associated with operations. The change-requirement vector is an N-tuple of attributes representing the states other attributes must be in, before the attribute can change state.

For each attribute there is a table of action names and times. The entry at row i, column j, of this table provides the action name and time to change the attribute from state i to state j. The times are initially obtained from a data base of standard times for maintenance actions.

As an example of how these vectors interact, consider a simple system in which a particular test can be performed only if the power is ON and the cover is SECURED. Suppose the cover state may be changed only when the power is OFF, and suppose that the system is currently turned ON, with the cover REMOVED. To perform the test, therefore, requires that the power be turned OFF (to allow the cover attribute to be changed), the cover be SECURED, and then the power turned back ON. The action-generating algorithm must obviously be capable of finding its way to a necessary condition, even if this involves temporarily moving away from the goal state.

To generate an action sequence enabling the performance of any operation, operation i, the model compares the current states of the system attributes to the states necessary to perform the operation. It forms a set of 'rational actions' which are those actions necessary to change the attributes which are currently not in their required states. It may be that some of these attribute state changes are themselves constrained. Consequently, the state changes necessary to enable the original state changes are added to the set of rational actions. This process is continued until all rational state changes in the set are expanded.

From this set, all actions which cannot be performed in the current system state are deleted, leaving a set of actions found to be both necessary and allowed. Each action in the rational set is now evaluated to identify the action which seems to best move toward the goal state (the state required by the operation under consideration) and <u>also enable the most state changes in the future</u>. The score, for any action i under consideration, is computed as follows:

$$V = NG - SE$$

where

V is the scored value of performing action i,

NG = the number of states not in their goal states, following performance of action i

and SE = the number of states which can be changed, following performance of action i.

The model selects for performance the action yielding the <u>lowest</u> V score, it adds the time of that action to the time to perform the operation being evaluated, and it updates the provisional system state vector.

During generation of an action sequence, a system state which was previously encountered in the sequence may be encountered again. In this case, the action leading to the duplicated state is discarded for one which leads to a new state, for the work leading from the first state to its duplicate is clearly nonproductive. If this successive substitution exhausts all possible actions at a decision point (i.e. a dead-end is encountered) the previous action in the sequence is replaced with an alternative. This process continues until an action sequence is produced which achieves the state required by the operation. The total operation time is then computed as the sum of the conditional time to perform the enabling actions plus the time to perform the fixed portion.

When all operation times have been computed in this manner, the model proceeds to determine their respective fault-isolation contributions, and finally to compute overall utilities.

Appendix A presents a detailed example of this action sequence generation.

Restrictions

The formulation described above is not the most general one possible. One assumption is that changing the state of one attribute does not change the state of any other attribute. For example, changing the power does not change the status of the cover. In addition, the state vectors representing the operation performance requirements and attribute change requirements are restricted to 'AND's, i.e. all designated states must be attained (whereas an OR function could specify that any one of the listed states could create the condition required). Finally, the attribute change vector is assumed to apply, regardless of the particular states involved. For example, the requirements to change the state of the cover are the same, whether the cover is to be SECURED or REMOVED.

These restrictions allow the computation of sequences to be accomplished in a reasonable amount of time, and they have not seriously degraded the efficiency of the sequences generated. It seems likely that human workers also restrict the search-space in generating action sequences, especially when the time consequences do not warrant detailed planning.

SECTION III. EXPERIMENTAL EVALUATION

The earlier two studies of corrective maintenance performance were highly controlled, to limit the sources of variation. The primary purpose of the study described below was to determine how closely the performance model could project actual maintenance performance with nearly all experimental controls removed. The study was conducted in an environment similar to a repair depot facility, with participants performing all the corrective maintenance work except for component replacement. Replacements were made by the experimenter, as the participants looked on. No controls were exerted to affect maintenance strategy, workpace, or the commission of errors.

A second aspect of the study was to employ the performance model, for the first time, to analyze the implications of some proposed design changes.

The experimentally obtained data provided detailed testing sequences for each participant, including performance times for all manual operations and inter-test cognitive times.

Experimental Method

Participants. Participants in the study were ten U.S. Navy electronics instructors from the Advanced Electronics School Department (AESD), San Diego, California. The technicians participated in the study voluntarily, and varied considerably in years and type of experience.

Maintenance Task. An infrared (IR) transmitter/receiver system was constructed to be easily transformed between two alternate design configurations. In the first configuration, design A, a number of design enhancements were provided to facilitate isolation of faults. In design B these special features were removed.

Each of the ten technicians was presented a sequence of eight corrective maintenance problems, inserted into the IR transmitter/receiver system. Four of the problems were presented in design A, and four were in design B. The task was to isolate the inserted fault by use of built-in indicators and peripheral test equipment.

The IR transmitter/receiver was constructed on five printed circuit-boards (see Appendix B). The function of the system was to transmit, via an infrared carrier, a two-digit number dialed into thumbwheel switches at the transmitter. The transmitter consisted of a 9-volt power supply, a digital section for encoding the dialed number into a serial bit-stream, and an IR transmitter section which performed frequency shifting of the carrier, which was then sent out over a fiber-optic cable via an infrared light-emitting diode. The receiver consisted of a 9-volt power supply, a section which decoded the modulated IR light beam into a digital signal, and a digital receiver section for converting the serial data to parallel form for display on two 7-segment LED displays. The system employed the following components:

Component Type	Quantity
integrated circuit (IC)	20
switch	2
transistor	2
cable	4
power supply	2
crystal	2
adjustable potentiometer	1
diode	1
light-emitting diode	4

The twenty IC's varied from 8-lead to 16-lead elements, producing a system of substantial complexity. In design B, the only built-in indication was a digital display of the number received and decoded by the receiver section. During normal operation this digital readout matched the settings of the thumbwheel switches at the transmitter.

Design A offered additional features to facilitate fault isolation. First, a cable was provided which could be used to connect the digital section of the transmitter directly to the digital section of the receiver, thereby bypassing the analog sections of the system. Secondly, two known-good circuit boards were provided which could be quickly swapped for their counterparts in the complete system. When used properly, the bypass cable and the two replacement boards allowed fault isolation to the circuit board level in just two steps, with minimal requirements for understanding the operation of the system. Finally, design A contained an additional built-in indicator, a two-color LED which shone green when the receiver was properly adjusted, and red otherwise (one of the faults introduced was a misadjustment of the receiver).

Experimental Procedure. Detailed technical documentation was provided each technician, including the theory of operation, schematic diagrams, and normal waveforms at the 56 test points. No troubleshooting hints were provided. Those instructions relating to the special features of design A explained the functions of the additional features, but offered no suggestions concerning application of those features.

Each technician studied the documentation at his own pace, and then viewed a 45-minute video tape which reviewed the theory of operation and demonstrated the normal waveforms throughout the system. The technician then worked a practice problem to become familiar with the system, followed by eight problems, four in each of the two configurations. The problem order and design assignment were counterbalanced to negate order effects. Appendix C provides the list of problems and the schedule of presentation.

At the start of each problem the participant left the room, and the experimenter inserted a faulty or misadjusted component into the IR system. The participant returned and was informed that the IR system was not operating properly. The technician performed all troubleshooting work except for component replacements, which were performed by the experimenter. A problem was terminated when the technician stated that the system was operational. Since system operation was an easy judgement to make, there were no cases of claiming completion when the fault was not actually resolved. There were, however, a number of cases where technicians continued troubleshooting after the actual fault had been replaced, as a result of errors in performing confirming tests.

A fixed, color/sound video camera recorded the performance of each technician. To ensure correct identification of test points, the experimenter stood nearby and recited into a lapel microphone the test point numbers being measured.

Data reduction was facilitated by a special-purpose computer program which controlled the playback of the video tapes, and allowed a viewer to mark the beginning and end of each maintenance operation as it was shown.

At the beginning of each problem, the viewer keyed in the technician number and problem number, and started the playback. Upon seeing the technician start to perform a manual operation, the viewer pressed a key, automatically capturing the frame number from the video tape at which the operation began (this did not halt playback). This frame number marked the end of a previous cognitive time interval, if any, and the beginning of a manual time interval.

When the viewer saw the operation completed, he pressed another key. The program then recorded the frame number of the operation termination, it automatically computed the elapsed time of the operation, and it marked the beginning of a cognitive time interval. The viewer then keyed in a code which identified the operation and resumed playback.

Results

Maintenance Times. Projected maintenance times were generated by running the performance model on data representating the two configurations of the infrared system. Ten projections of manual performance times were obtained for each problem, in each configuration, by running the model in the sampling mode. The spares availability parameter was set to the same value obtained from earlier studies, as was the urgency parameter.

The mean predicted manual time across problems for design A was 443 seconds, while for design B it was 517 seconds. This difference was considerably less than expected. Intuitively, design A seemed to offer substantial improvements.

The mean actual manual maintenance time, across problems, was 395 seconds for design A, and 327 seconds for design B. A one-way ANOVA showed this 68-second difference between the two designs to be non-significant ($p>0.4$). For design A, the correlation between PROFILE projections and subject data was $r=0.89$ (Fisher $Z=3.15$, $p<.001$); for design B, $r=0.77$ ($Z=2.26$, $p<.01$).

Distributions of manual maintenance times for the two designs, across problems, are shown in Figures 2 and 3, along with the actual means and projected means. These figures illustrate the small difference between the two designs, in either actual or projected data, compared to the variance of the distributions.

Tables 1 and 2 present the actual and projected times for each problem, for designs A and B, respectively.

		Actual Data		Projection	
Problem	N	Mean	S.D.	Mean	S.D.
1	6	692	539	845	150
2	6	307	164	234	52
3	5	335	220	254	107
4	6	448	168	880	80
5	5	334	104	131	62
6	5	602	303	856	87
7	5	217	112	169	16
8	5	170	126	46	45
Mean		395	217	443	77

Table 1. Actual and Projected Manual Maintenance Times, Design A. (times in seconds)

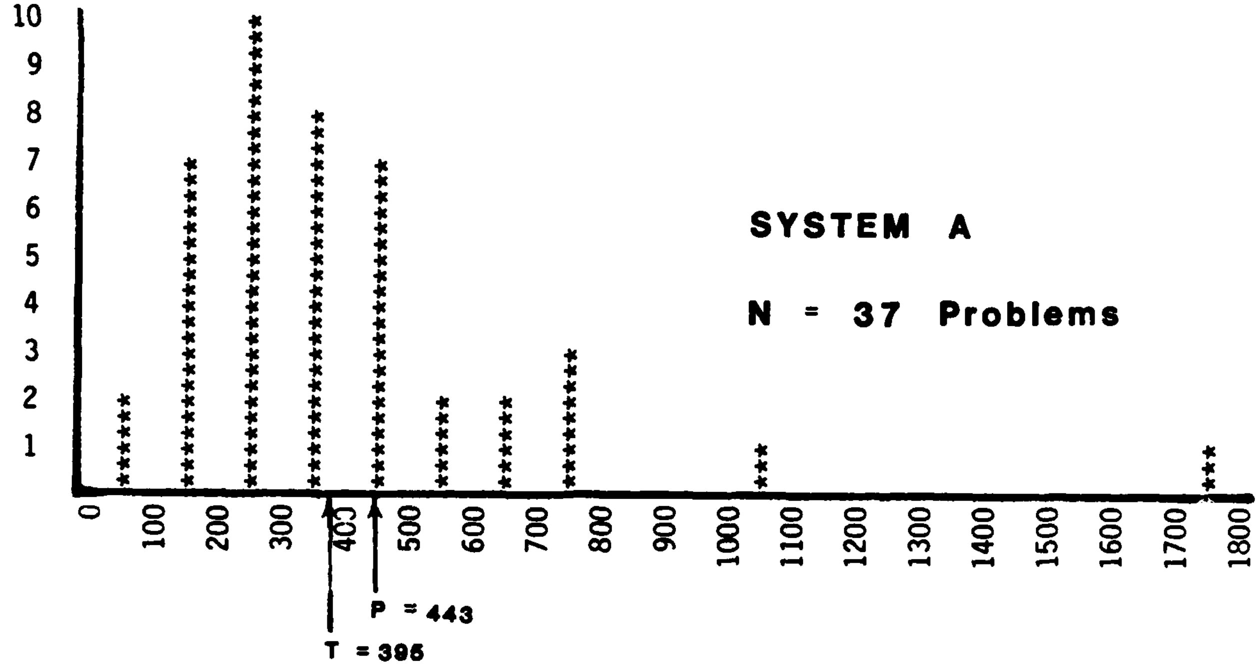


Figure 2. Actual manual maintenance times (sec.). SYSTEM A

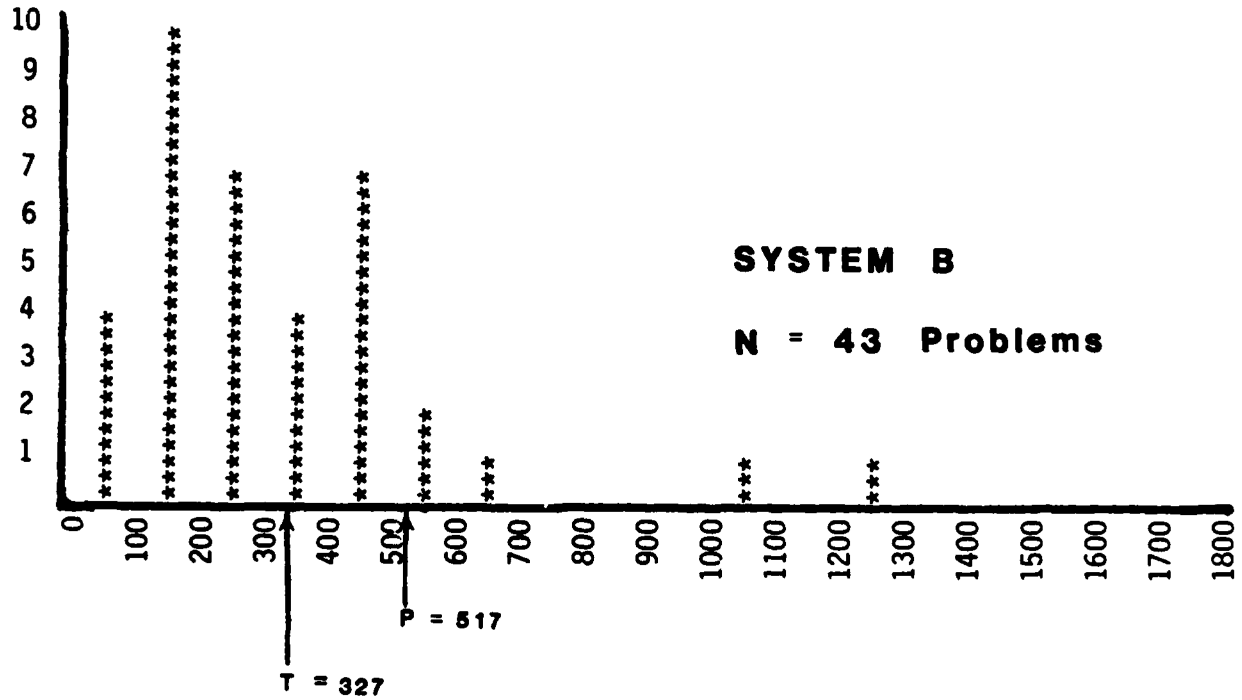


Figure 3. Actual manual maintenance times (sec.). SYSTEM B

* ONE PROBLEM BY NAVY TECHNICIAN

T MEAN OF TECHNICIAN DISTRIBUTION

P MEAN OF PROFILE PROJECTION

Problem	N	Actual Data Mean	Actual Data Std.	Projection Mean	Projection Std.
1	4	564	340	825	100
2	5	242	110	669	192
3	5	367	210	583	77
4	4	415	108	897	65
5	5	190	78	155	47
6	4	546	462	838	57
7	5	111	46	254	38
8	5	293	94	116	96
Mean		327	181	517	84

Table 2. Actual and Projected Manual Maintenance Times, Design B. (times in seconds)

The PROFILE analysis projected a 74-second performance improvement for design A, which is small in comparison to the means themselves. Several explanations are possible for why this projected improvement was not realized. The most likely one, of course, is that the technique cannot reliably make such fine discriminations, especially for such small samples.

The other effect worth noting is that some of the ten participating technicians did not employ the 'improved' design features to an appreciable extent until they encountered difficulty in isolating a fault (recall that most of the aids were elective features which could be put into service by choice). It may be that these participants regarded swapping and patching techniques as less effective or legitimate than the use of conventional test equipment.

Work Content. The data presented above pertain to the ability of the performance model to predict the time to isolate and repair faults. Another important consideration is the success with which the performance model projects the work performed. To make this evaluation, all the manual work was classified into fourteen categories, as shown in Table 3.

Category	Work Content
1	Read a static voltage value
2	Check a digital control or clock pulse
3	Read a data value - Low-level analog
4	Read a data value - FM signal
5	Read a data value - Digital
6	Check receiver read-out and transmitter setting
7	Check phase-lock loop indicator
8	Adjust phase-lock loop
9	Replace a DIP component
10	Replace a cable
11	Replace a power supply
12	Swap a known-good printed-circuit board for suspected board
13	Delete section with bypass cable
14	Replace a soldered component

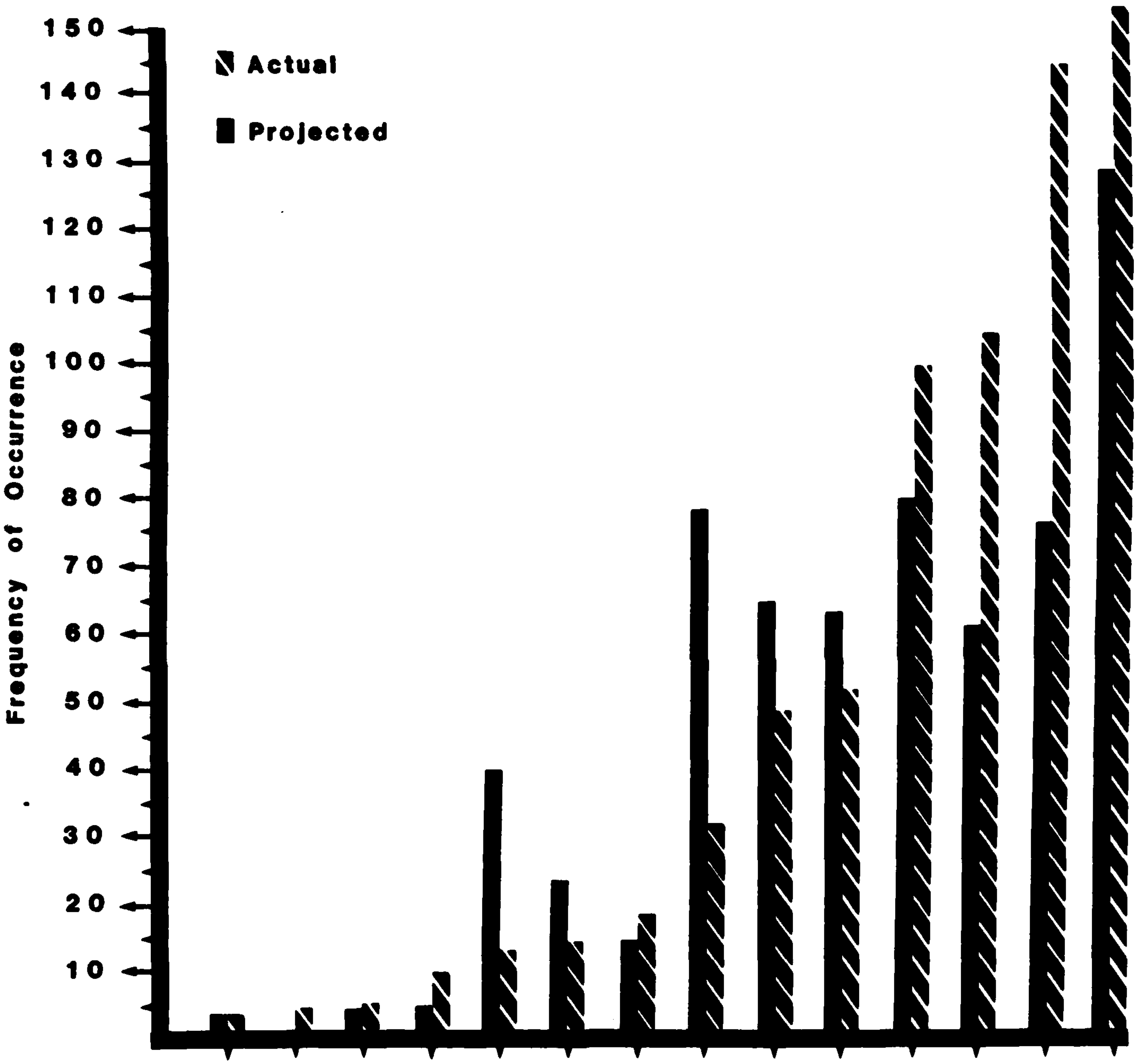
Table 3. Fourteen Work Categories for Maintaining the IR System.

Figure 4 presents the actual frequencies with which each work category was performed, over 43 problems in design A, and projected frequencies for the same number of problems. A Chi-square test indicates that the differences between projected times and actual times are statistically significant.

Individual Differences. Table 4 presents the average manual and inter-test cognitive times per problem, by technician. Inter-test cognitive time is the time during which no observable manual work is performed.

Technician Number	Manual Time	Cognitive Time	Total Time	Cognitive/Total
1	347	230	577	.40
2	451	320	771	.42
3	375	251	626	.40
4	566	540	1106	.49
5	241	363	604	.60
6	407	430	837	.51
7	501	376	877	.43
8	282	391	673	.58
9	282	476	758	.63
10	199	476	675	.71
Mean	365	385	750	.51

Table 4. Average Manual and Inter-test Cognitive Times, by Technician.



CATAGORY	12	14	11	10	13	8	3	1	9	7	4	6	5	2
ACTUAL	3	4	5	9	13	14	18	32	48	52	100	114	144	152
PROJECTED	3	0	4	4	40	23	14	78	65	63	80	61	76	128

Figure 4. Actual and Projected Work Content, Design A

While the sample is not large, these results pertaining to technician variability conform closely to those obtained in earlier studies, in that the time of the slowest worker was only about twice (1.9) the time of the fastest worker. Also, as we have seen with other hardware systems and technician samples, the variations attributable to the nature of the fault are considerably greater than those attributable to variations in individual performance, within a homogeneous group. In this study, the most time-consuming problem was not unusually difficult, yet it was approximately four times as lengthy as the easiest problem.

Also, the variation over technicians in the portion of time allocated to cognitive work was relatively small, ranging from .40 to .71. The ratio was even more consistent by problem, ranging from .41 for problem six, to .59 for problem five.

As the ten participants were all Navy instructors, we might assume that the sample is more homogeneous than would be encountered in the field (although their years of experience varied from five to twenty-eight). Still, the results are most encouraging, as they suggest that variations in individual performance will not mask out the effects of design on maintainability. This is not to say that design factors are more, or less, critical than personnel factors such as selection, motivation, and training. Rather, it supports the notion that a design can be evaluated in the context of some defined maintainer population.

SECTION IV. MAINTENANCE COMPLEXITY

Definition

Considerable research has been devoted to examining what design features cause systems to be complex, and how increased complexity affects maintainability (Wohl, 1980; Rouse and Rouse, 1979; Nauta and Bragg, 1980; DePuy (ed.), 1982). A prerequisite to such considerations is some statement concerning what effects are meant by the word 'complex'. For the purpose of evaluating system maintainability, a promising approach is to define complexity in terms of the difficulty of isolating faults, as measured by the number of indicators employed to isolate various malfunctions in the system. To emphasize the intent of this complexity measure, we term it Maintenance Complexity, or MC.

An 'indicator', by our terminology, is an element which reflects some single characteristic of a system's function. Examples are panel lights, meters, audio tones, and even smells or vibrations. The model's fault-effects matrix associates replaceable units to symptoms which could be produced by those units.

Tests, on the other hand, are operations which sample one or more indicators. A computer 'boot-strap', or start-up, test might produce symptom information from a disk-access light, a CRT display, and a disk motor sound. Since tests vary considerably in the number of information sources they monitor, they are not the ideal unit of measure for computing maintenance complexity.

The computation of Maintenance Complexity, MC, for a single fault, is as follows:

$$MC = I + C$$

where MC is Maintenance Complexity, for the fault,

I = the number of different indicators employed by the performance model to isolate the fault,

C = the number of confirming tests, performed following replacements or adjustments, which repeat previously performed tests.

The value C is the number of <u>tests</u> which are repeated following adjustments or replacements (which is the only time tests are repeated by the performance model). The number of indicators involved in these tests is disregarded, since only one indicator which was previously abnormal needs to be monitored by the maintainer. If that indicator is abnormal when the test is repeated, then the fault was not resolved by the replacement or adjustment; if it is normal, then the fault has been resolved (assuming single faults).

A distribution of MC values, over an appropriate sample of faults, yields a mean complexity value, as well as indications of maximum complexity and variance in complexities. All of these could be useful measures in evaluating design for maintainability.

<u>Examples</u>

Table 5 presents mean MC for each of the four systems which have been analyzed by the model. The system with the lowest MC, at 4.8, is the 'block diagram' system (Appendix D) consisting of 17 replaceable elements, each exhibiting just one failure mode, with no feedback. The fault-effects data for this system reflects its simplicity; each fault's symptom pattern is unique and consists entirely of 0's (no effect) and/or 1's (single, certain effect).

Substantially more complex, at 11.1, is the computer system (Appendix E) consisting of 19 replaceable elements and considerable feedback. In this system, not all faults can be identified with unique symptom patterns, and many of the replaceable elements can fail in multiple modes. The infrared transmitter/receiver, consisting of 39 replaceable elements with multiple failure modes, is slightly more complex.

System	MC
Block diagram system	4.8
Computer	11.1
Infrared transmitter/receiver (design A)	12.5
Infrared transmitter/receiver (design B)	12.9

Table 5. Maintenance Complexities for Four Systems.

Characteristics Measured

One advantage of the MC computation is that it does not simply reflect system size. A very large computer with many components and circuit nodes, could yield a low MC value if its faults could be isolated with relatively few indicators. The MC computation is also not affected by the performance time of tests, which may bear no relationship to what we commonly mean by complexity.

The sparse data we have indicate a possible relationship between MC and cognitive time. As shown in Table 6 below, the cognitive time per step, and cognitive time per problem, are generally higher for the systems with higher MC values.

System	MC	Cognitive Time (sec.) per Problem	per Step
Block diagram system	4.8	135	18.5
Computer	11.1	265	35.1
Infrared transmitter/receiver (design A)	12.5	467	24.3
Infrared transmitter/receiver (design B)	12.9	393	24.7

Table 6. Maintenance Complexities and Cognitive Times.

A consistent relationship between MC and cognitive time per problem would not be particularly surprising, as MC is a measure of problem length, and cognitive time increases as the number of tests increase. If MC and cognitive time per step are reliably related, however, then MC may have more general application in projecting cognitive workload.

Application in Design

Numerous options could be pursued by a system designer to reduce system complexity. Indicators might be selected or devised which provide more discriminating (less confounded) symptoms, thereby reducing the number of indicators sampled. A similar effect might be attained by revising the design of the replaceable units so that elements are more functionally allocated.

A more involved design option is to reduce the number of indicators, perhaps by adding circuitry to evaluate multiple functions, and to display a simple binary indication of system status. Naturally, such additions could increase manufacturing costs and weight, and could decrease reliability. The designer would have to consider these factors in relation to the expected maintenance improvements. The MC measure, though, would sense simplifications in maintenance testing achievable by increased hardware, whereas most other techniques would view the addition of any circuitry as necessarily increasing complexity, even if fault-isolation time and effort were thereby reduced.

SECTION V. SUMMARY AND CONCLUSIONS

The study described in Section III, and those conducted previously, provide information concerning the effectiveness of the maintenance performance model in meeting two, somewhat different, maintainability prediction needs, 1) projection of total expected repair time distributions, and 2) evaluation of design variables as they impact maintainability.

Projecting Maintenance Time Distributions

As a tool for making accurate predictions of total repair time, the model seems to be well-founded, but in need of additional capabilities. The primary facilities which are lacking are 1) the capability to predict the commission and consequences of manual and perceptual performance errors, 2) a means for quantifying the perceived impact upon the maintainer of environmental factors such as danger and discomfort, and 3) a quantitative basis for projecting cognitive time.

Of these three factors, the projection of human error would seem to present the greatest difficulty. Both error commission and error detection are formidable variables affecting performance.

Projecting mean cognitive times may now be feasible, if the tentative findings of the maintenance studies prove reliable. An earlier study (Towne, Johnson, and Corwin, 1982) showed that mean inter-step (inter-test) cognitive times within a system increase with the mean manual performance times of the tests available. This is consistent with a productivity-maximization mode of performance, as increased decision time is warranted when the time consequence are greater. The ratios of cognitive time to manual time, within each of the four systems now studied, are remarkably consistent across problems and individual technicians, supporting the notion that cognitive effort may be primarily related to general complexity of the system and the magnitude of the manual effort anticipated at each step of a problem.

Exploring Design as it Affects Maintainability

As a means for exploring the effects of design upon maintainability, the model is more complete. In such an application, some baseline of maintainer capability and maintenance environment would be desirable. A capable maintainer working in a non-hazardous environment may be an acceptable condition for conducting analysis of equipment design.

Nature of the Task Sequences. An earlier section stressed that the performance model employs a simple productivity rule for selecting maintenance operations, rather than attempting to model general human decision-making processes, or to apply an individual expert's rules of behavior. The model was developed to produce efficient corrective maintenance performance, utilizing whatever algorithmic approaches would accomplish that objective.

While the original model was broadened to consider consumption of spares in relation to the urgency of the maintenance requirement, it has not been augmented with special rules or processes to handle the seemingly endless special situations which corrective maintenance presents. The model's testing sequences result instead from the application of a simple performance criterion to a rich and complex data pool. These data include both the system-knowledge (fault effects, reliabilities, costs, accessibility) and data about the particular problem underway (symptoms received and equipment conditions established).

When applied to these varied data types, the simple optimization criterion can produce testing sequences exhibiting a number of characteristics which are not explicitly formulated in the model. These include focusing, hypothesis testing, continuity, and opportunistic decision-making.

For each possible operation considered by the model during a problem, computation of the two required ingredients, expected information value and time to accomplish, is time-consuming. This computational load has been mitigated by the implementation of a number of restrictions, heuristics, and sub-optimization algorithms which seem to produce reasonable approximations to the true optimums.

Future Plans

Considerable research could, and should, be conducted to further understand how human workpace, error commission rates, and fault isolation operation sequences are affected by the work environment. Variables of interest would include urgency of the situation, danger and discomfort of alternative operations, and the complexity of the system, in relation to the experience and training of the worker. In addition, the tentative findings relating cognitive time to manual performance times and to Maintenance Complexity should be explored in detail. This line of research would provide many of the ingredients necessary to project total corrective maintenance times.

Our work in the coming year will be concerned with investigating the feasibility of embedding the maintenance performance model within a computer-driven, graphic design aid. The primary objective is to determine the extent to which a Computer-Aided Design (CAD) system can provide intelligent design assistance in the maintainability domain. An important prerequisite is the marrying of the CAD data representations with those of the performance model, so that the model's data needs are provided, as much as possible, by system design representations acquired by the normal CAD process.

References

DePuy, W.D. (ed.) Report of the defense science board 1981 summer study panel on operational readiness with high performance systems, April 1982, Washington D.C.

Nauta, F. & Bragg, L. Army electronic equipment support options: Assessment of maintenance complexity. November 1980, Logistics Management Institute, Washington, D.C.

Rouse, W.B. & Rouse, S.H. Measures of the complexity of fault diagnosis tasks. IEEE Transactions on Systems Man and Cybernetics, 1979, (11), 720-727.

Towne, D.M., Fehling, M.R., & Bond, N.A. Design for the maintainer: projecting maintenance performance from design characteristics. Behavioral Technology Laboratories, Technical Report No. 95, July 1981, University of Southern California.

Towne, D.M., Johnson, M.C., & Corwin, W.H. PROFILE: A technique for projecting maintenance performance from design characteristics. Behavioral Technology Laboratories, Technical Report No. 100, November 1982, University of Southern California.

Wohl, J.G. Diagnostic behavior, system complexity, and repair time: a predictive theory (Technical Report M80-0008). Bedford, Massachusetts: MITRE Corporation, April, 1980.

APPENDIX A

CONDITIONAL TASK SEQUENCE GENERATION

The following is an example of the conditional time computation. The example system is a letter-quality printer. It has a power switch and a print density switch on the outside of the case, and a lid which can be raised to expose the print mechanism. The print mechanism consists of a ribbon and type face wheel, both of which can be changed.

Given that the printer is in some state, the conditional algorithm will produce the actions which must be performed to put the printer in some other state required by some operation, such as a test or replacement. In this discussion, an 'attribute' is a facet of the system that can change state, such as the power status of the printer (ON <---> OFF). The 'goal state' is a list of attribute states which are required by a test. Note that not all attributes of the system need be specified for a given test. If an attribute has no effect on a test it is omitted from the 'goal state' list for that test.

In a trivial example where all attributes of a system are context-free, (i.e. free to change at any time), the algorithm could merely change attributes as needed to move the system state to the goal state. Here, we will say that the power switch can be moved to OFF or ON, and the print density can be changed at any time. However, in this as in many other systems, dependencies exist where certain attributes cannot change state unless other attributes are in some required state. In such a case, the algorithm must perform certain steps (state changes) before it can move the system to the goal state. For example, the algorithm must 'open' the lid to gain access to the print wheel, in order to change it.

Below is a table describing the printer system.

PRINTER SYSTEM

	Possible States		
Attribute	1	2	3
1: Power	on	off	
2: Lid	closed	open	
3: Ribbon	fabric	carbon	removed
4: Character Wheel	Pica	Elite	
5: Print Density	Ten/inch	Twelve/inch	

State Change	System Requirements to Enable State Change	
1: Power	-	(no requirements)
2: Lid	1(2)	(power must be off)
3: Ribbon	2(2)	(lid must be open)
4: Character Wheel	3(3)	(ribbon must be removed)
5: Print Density	-	(no requirements)

Suppose the system is initially in this state:

Attribute	State	
Power	1	(On)
Lid	1	(Closed)
Ribbon	1	(Cloth)
Char. Wheel	1	(Pica)
Print Density	1	(10 characters/Inch)

Below is a demonstration of the algorithm which performs the actions to change the ribbon type, the character wheel type, and the print density. The goal state is as follows:

Attribute	State	
Power	1	(On)
Lid	1	(Closed)
Ribbon	2	(Carbon)
Char. Wheel	2	(Elite)
Print Density	2	(12 characters/Inch)

The initial system state can be represented as: 1(1) 2(1) 3(1) 4(1) 5(1)
The required state can be represented as: 1(1) 2(1) 3(2) 4(2) 5(2)

PROCEDURE:

STATE 1: 1(1) 2(1) 3(1) 4(1) 5(1).
Determine rational action list:
Test requires: 3(2) 4(2) 5(2) (goal states yet to be met)
3(2) requires 2(2).
2(2) requires 1(2).

Provisional Rational actions: 1(2) 2(2) 3(2) 4(2) 5(2).

But, 2(2), 3(2), 4(2) cannot be changed in the current system state, so they are deleted as possible actions.

Final Rational actions: 1(2) 5(2).

Score each rational action:

1(2): new state: 1(2) 2(1) 3(1) 4(1) 5(1)
    # states not in goal state: 4
  - # states can be changed : <u>2</u>
            score : 2

5(2): new state: 1(1) 2(1) 3(1) 4(1) 5(2)
    # states not in goal state: 2
  - # states can be changed : <u>0</u>
            score : 2

Score is tied, so arbitrarily do 1(2). (POWER: off)

NEW STATE 2: 1(2) 2(1) 3(1) 4(1) 5(1).
Determine rational action list:
Test requires: 1(1) 3(2) 4(2) 5(2) (goal states yet to be met)
3(2) requires 2(2).
2(2) requires 1(2).
Provisional rational actions: 1(1),1(2) 2(2) 3(2) 4(2) 5(2).

1(2) is non-productive.
3(2), 4(2) cannot be changed in current system state.

Final rational actions: 1(1) 2(2) 5(2).
Score each rational action:
1(1): new state: 1(1) 2(1) 3(1) 4(1) 5(1)
This state was previously generated with fewer steps;
delete this as a possible action.

2(2): new state: 1(2) 2(2) 3(1) 4(1) 5(1).
    # states not in goal state: 5
    # states can be changed : <u>4</u>
            score : 1

5(2): new state: 1(2) 2(1) 3(1) 4(1) 5(2).
    # states not in goal state: 3
    # states can be changed : <u>1</u>
            score : 2

Best action is 2(2). (LID: Open)

NEW STATE 3: 1(2) 2(2) 3(1) 4(1) 5(1)
Rational Actions: 1(1) 2(1) 3(2) 3(3) 5(2) (computation details omitted)
1(1): score: 4 - 2 = 2
2(1): score: = infinity (resulting state previously encountered)
3(2): score: 4 - 3 = 1
3(3): score: 4 - 4 = 0
5(2): score: 4 - 3 = 1

Best action: 3(3) (RIBBON: Removed)

... system is ...

Attribute

Power
Lid
Ribbon
Char. Wheel
Print Density

Below is a demonstratio
to change the ribbon type, t
The goal state is as follows

Attribute

Power
Lid
Ribbon
Char. Wheel
Print Density

The initial system state can b
The required state can be repr

PROCEDURE:

STATE 1: 1(1) 2(1) 3(1) 4
Determine rational action list:
Test requires: 3(2) 4(2) 5(2)
3(2) requires 2(
2(2) requires 1(

Provisional Rational actions: 1(

But, 2(2), 3(2), 4(2) canno
so they are deleted as possible a

Final Rational actions: 1(2) 5(2

NEW STATE 4: 1(2) 2(2) 3(3) 4(1) 5(1)
Rational Actions: 1(1) 2(1) 3(2) 4(2) 5(2)
1(1): score: 3 - 2 = 1
2(1): score: 3 - 2 = 1
3(2): score: 3 - 2 = 1
4(2): score: 3 - 3 = 0
5(2): score: 3 - 3 = 0

4(2) and 5(2) tied, so chose 4(2) arbitrari

NEW STATE 5: 1(2) 2(2) 3(3) 4(2) 5(1)
Rational Actions: 1(1) 2(1) 3(2) 5(2)
1(1): score: 3 - 2 = 1
2(1): score: 3 - 2 = 1
3(2): score: 3 - 3 = 0
5(2): score: 3 - 3 = 0

3(2) and 5(2) tied, so chose 3(2) arbitrari

NEW STATE 6: 1(2) 2(2) 3(2) 4(2) 5(1)
Rational Actions: 1(1) 2(1) 5(2)
1(1): score: 3 - 2 = 1
2(1): score: 3 - 3 = 0
3(1): score: 3 - 3 = 0

2(1) and 3(1) tied, so chose 2(1) arbitrari

NEW STATE 7: 1(2) 2(1) 3(2) 4(2) 5(1)
Rational Actions: 1(1) 5(2)
1(1): score: 2 - 2 = 0
5(2): score: 2 - 2 = 0

1(1) and 5(2) tied, so chose 1(1) arbitrari

NEW STATE 8: 1(1) 2(1) 3(2) 4(2) 5(1)
Rational Actions: 5(2)

Best action is 5(2). (PRINT DENSITY: 12 Ch

NEW STATE 9: 1(1) 2(1) 3(2) 4(2) 5(2)
GOAL STATE: finished.

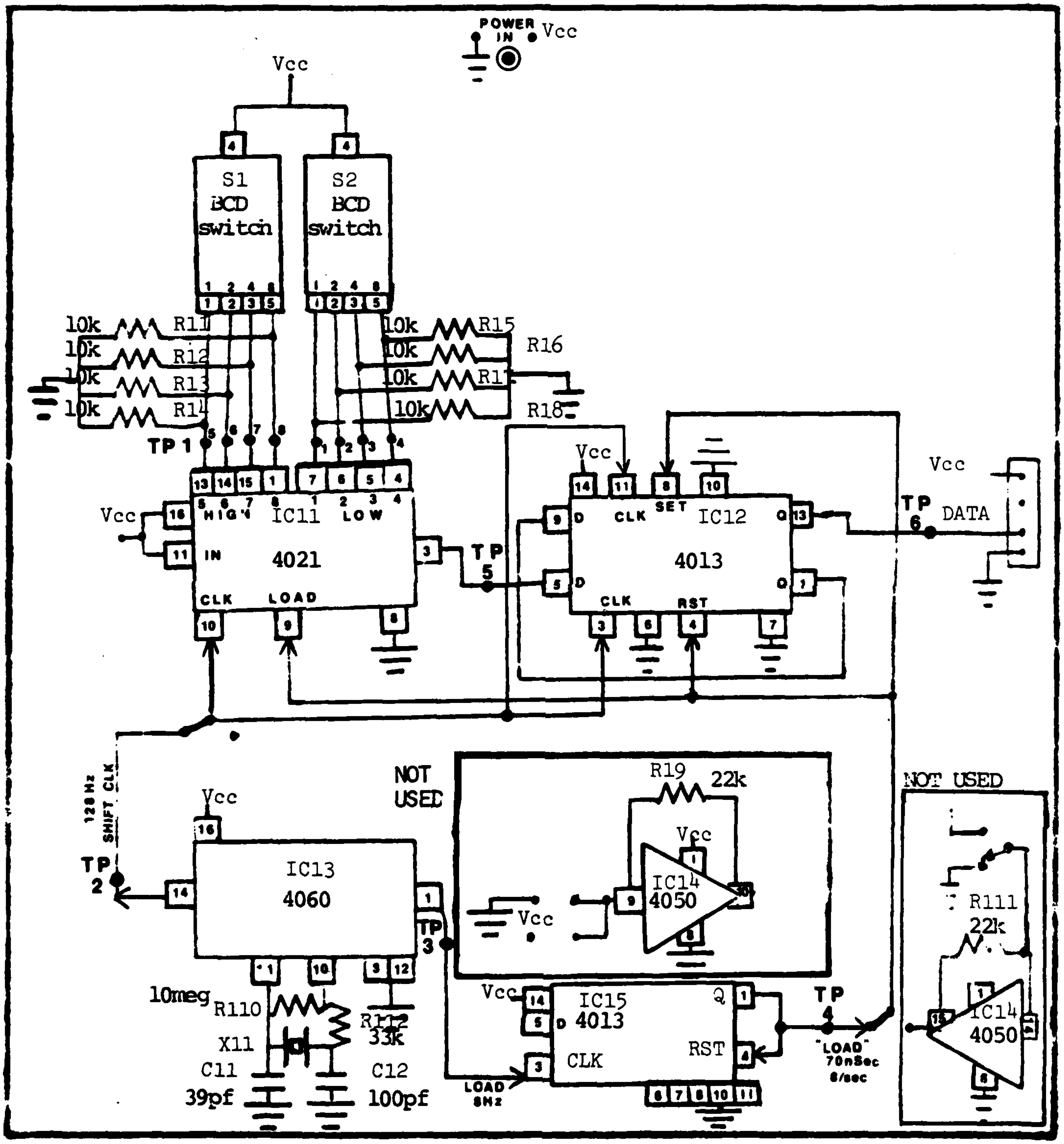
State transition sequence summary:

POWER:	Off
LID:	Open
RIBBON:	Removed
CHARACTER WHEEL:	Elite
RIBBON:	Carbon
LID:	Closed
POWER:	On

Discussion

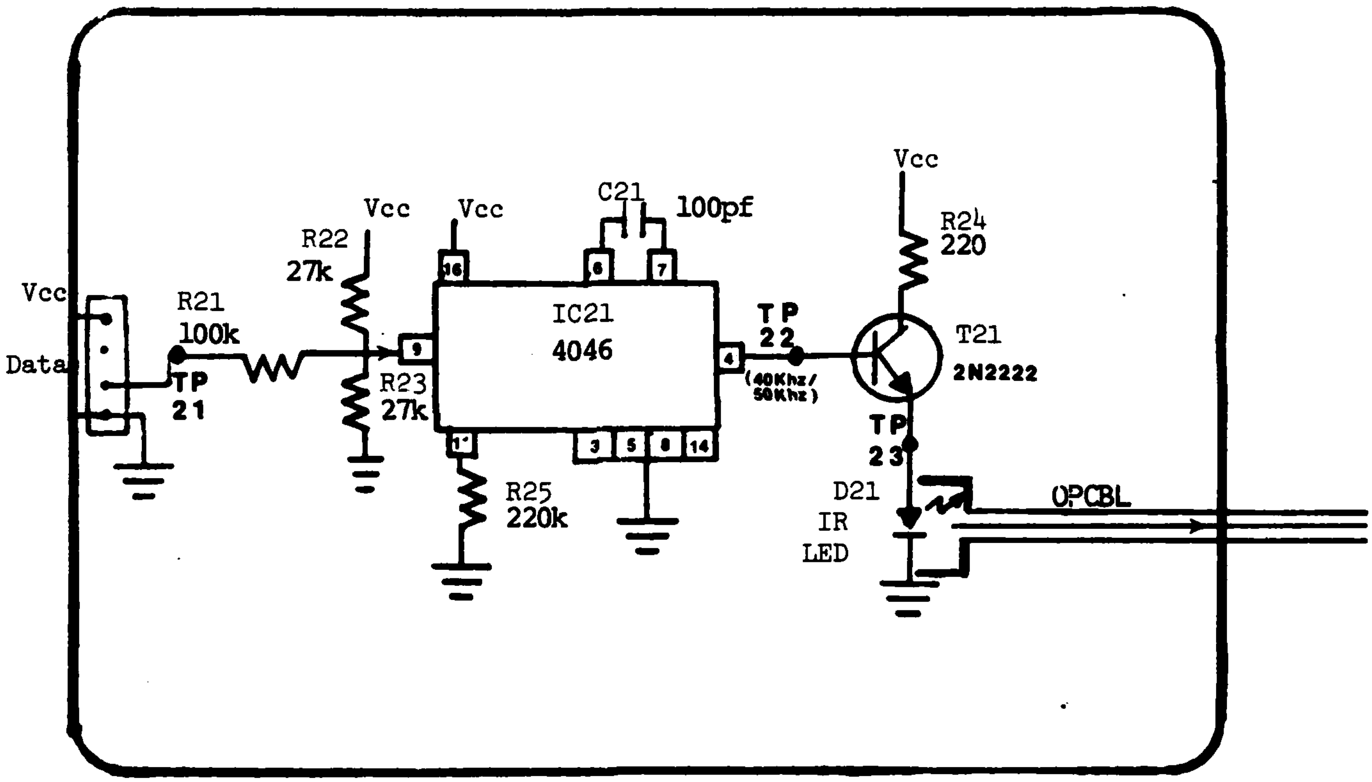
In this case, the algorithm achieved the goal state with its first choice at each step. It is possible, though, for a low scored rational action ("best action") at step "i" to lead to a dead end in which the only possible actions produce states which were previously encountered. In such a case, the algorithm trys the next best action, at step "i". If all actions at step "i" lead to dead ends, the algorithm backs up to the previous step "i-1", and trys the next best action at step "i-1", and so on, until finally some action leads to the goal state.

Also note that this does not attempt to produce an optimum step sequence. Rather, any successful solution is accepted. The heuristic scoring rule will produce an optimum provided the optimum does not require greater than one-step look-ahead, and in general will produce reasonable sequences. The algorithm may be modified to find the best sequence by enumerating all rational actions at each step. The compute load, however, would be severe.



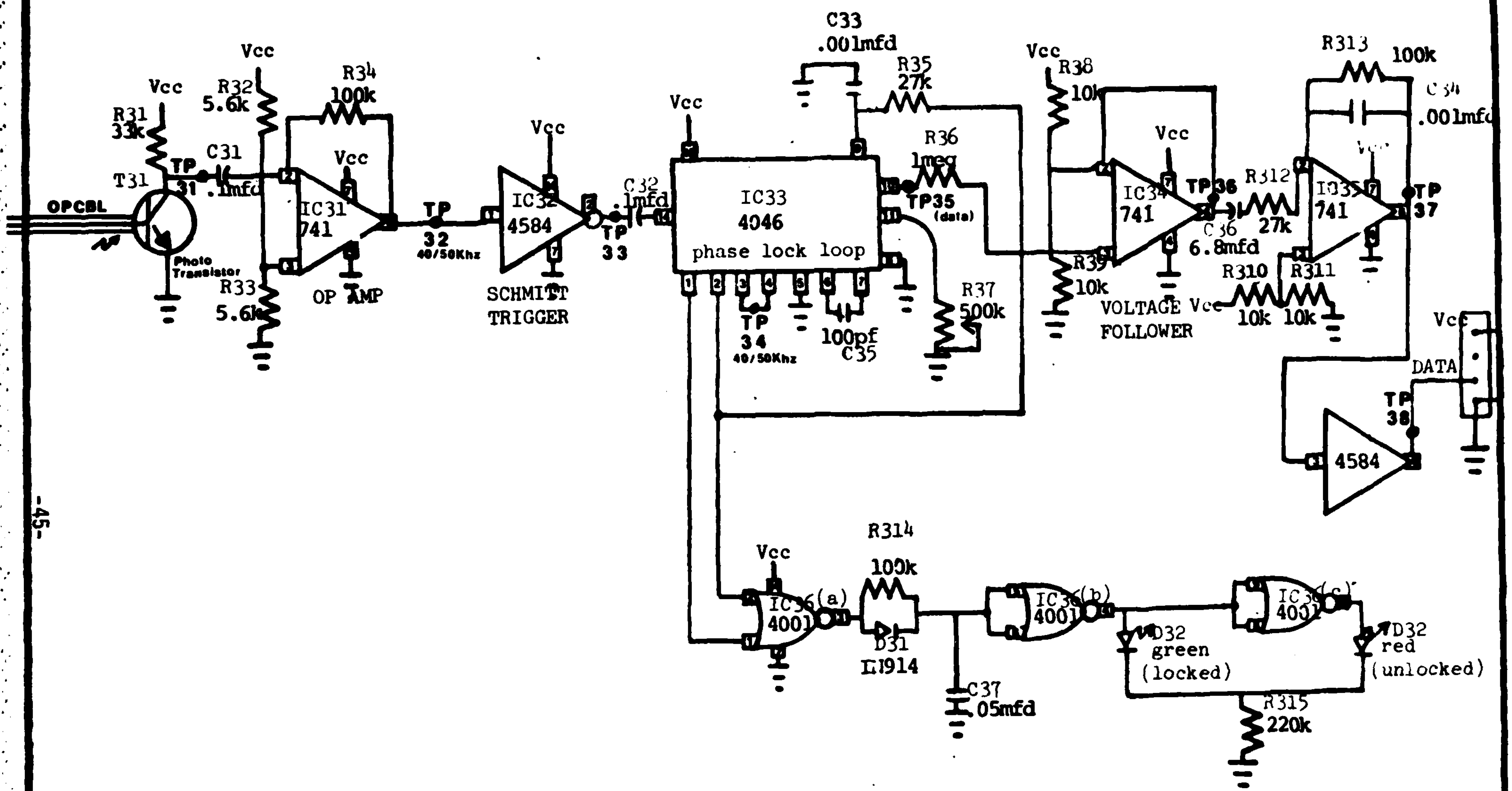
DIGITAL TRANSMITTER

Schematic 1



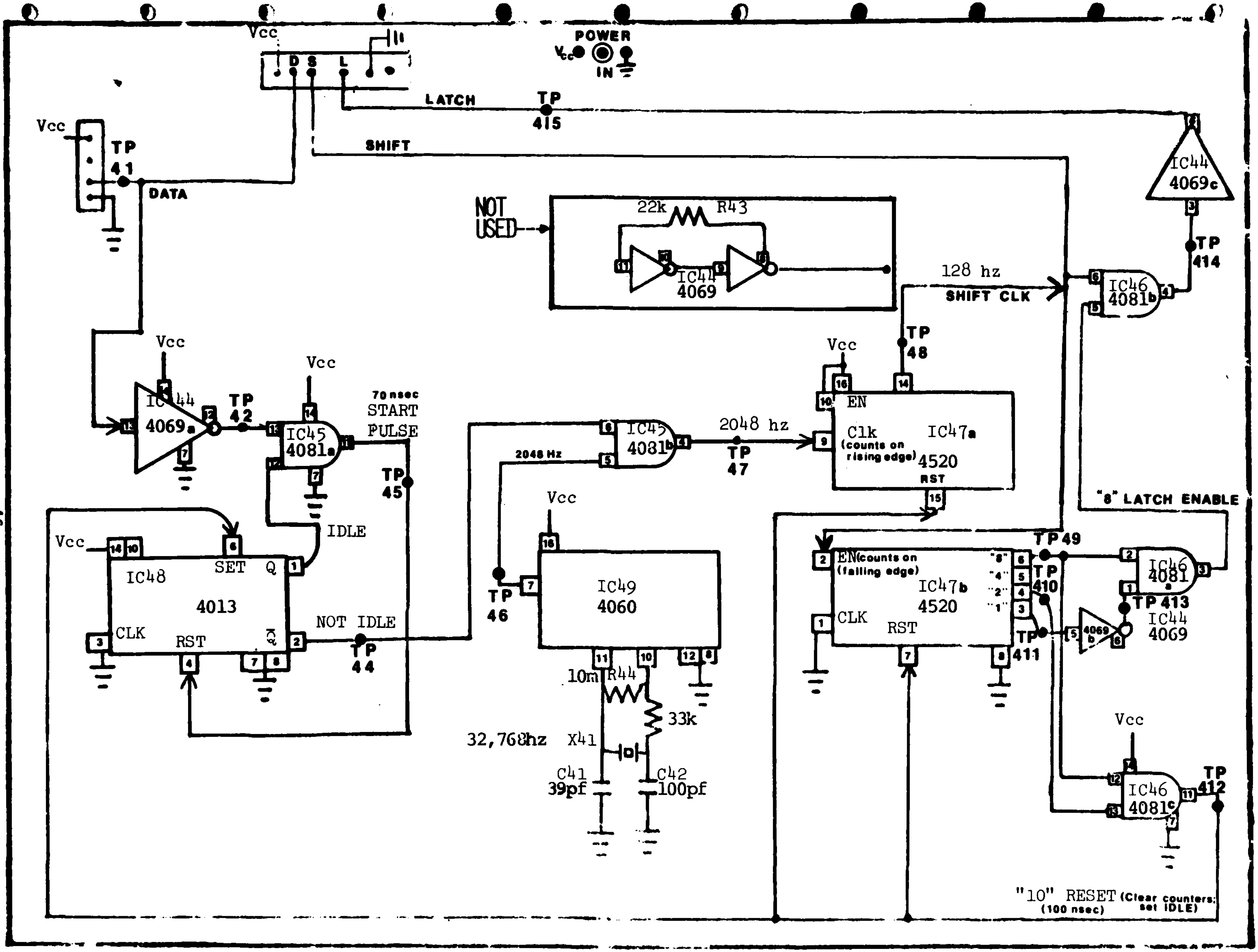
INFRA-RED TRANSMITTER

Schematic 2

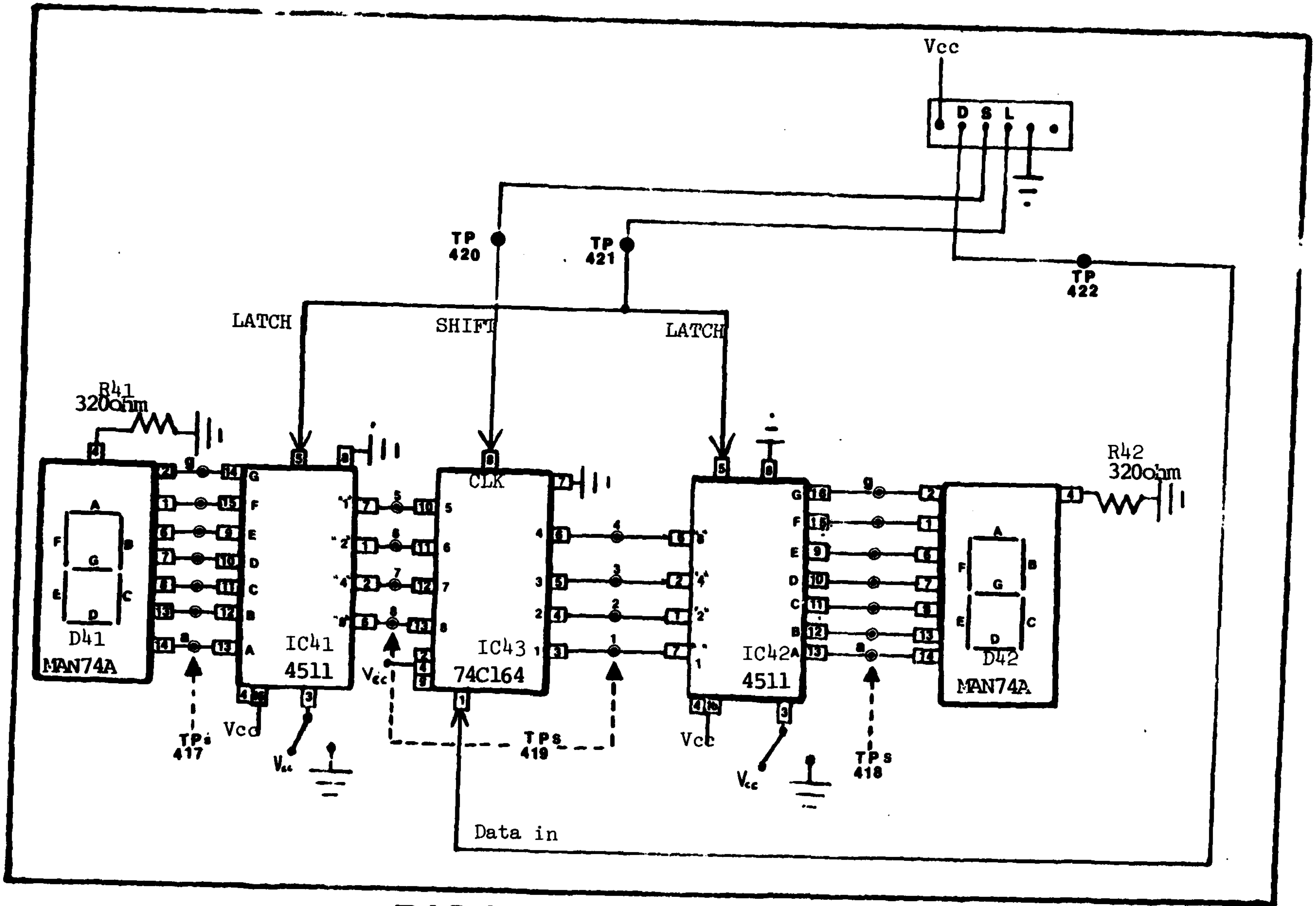


INFRA-RED RECEIVER

Schematic 3



Schematic 4a DIGITAL RECEIVER (LOGIC MODULE)



Schematic 4b DIGITAL RECEIVER

(DISPLAY MODULE)

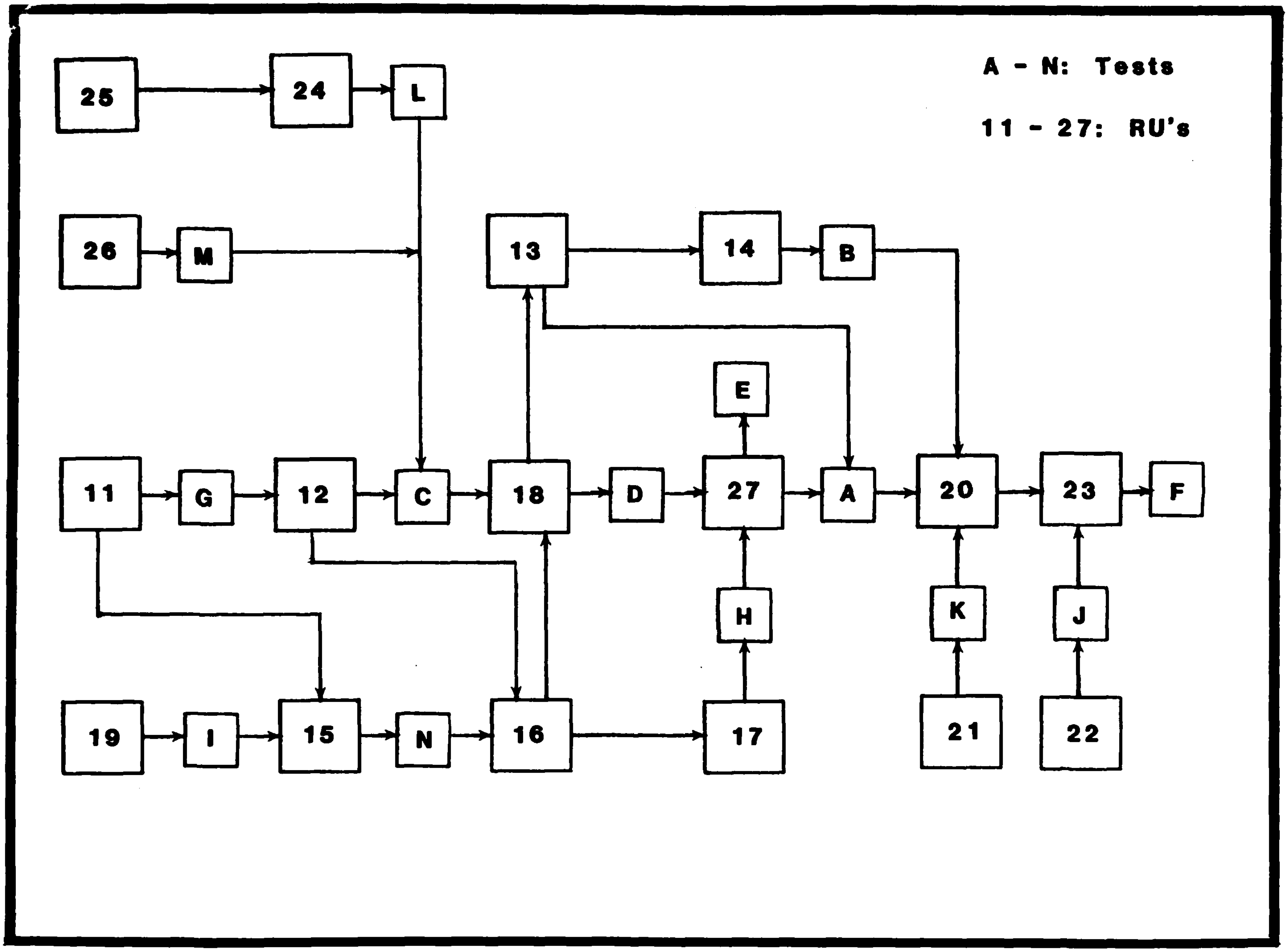
APPENDIX C

PROBLEMS, AND PROBLEM SEQUENCE ORDER
(Infrared Transmitter/Receiver)

Problem	RU number	Description	
0 (practice)	7	IC 21	4046 PLL
1	36	CBL3	Cable to Digital Display
2	10	IC 31	741 Op-Amp
3	14	IC 33	4046 PLL
4	27	IC 48	4013 Dual-D flip-flop
5	20	IC 41	4511 BCD latch
6	25	IC 46	4081 2-input AND
7	32	TPS	Transmitter 9-volt power supply
8	17 (adj.)	R37	500K potentiometer

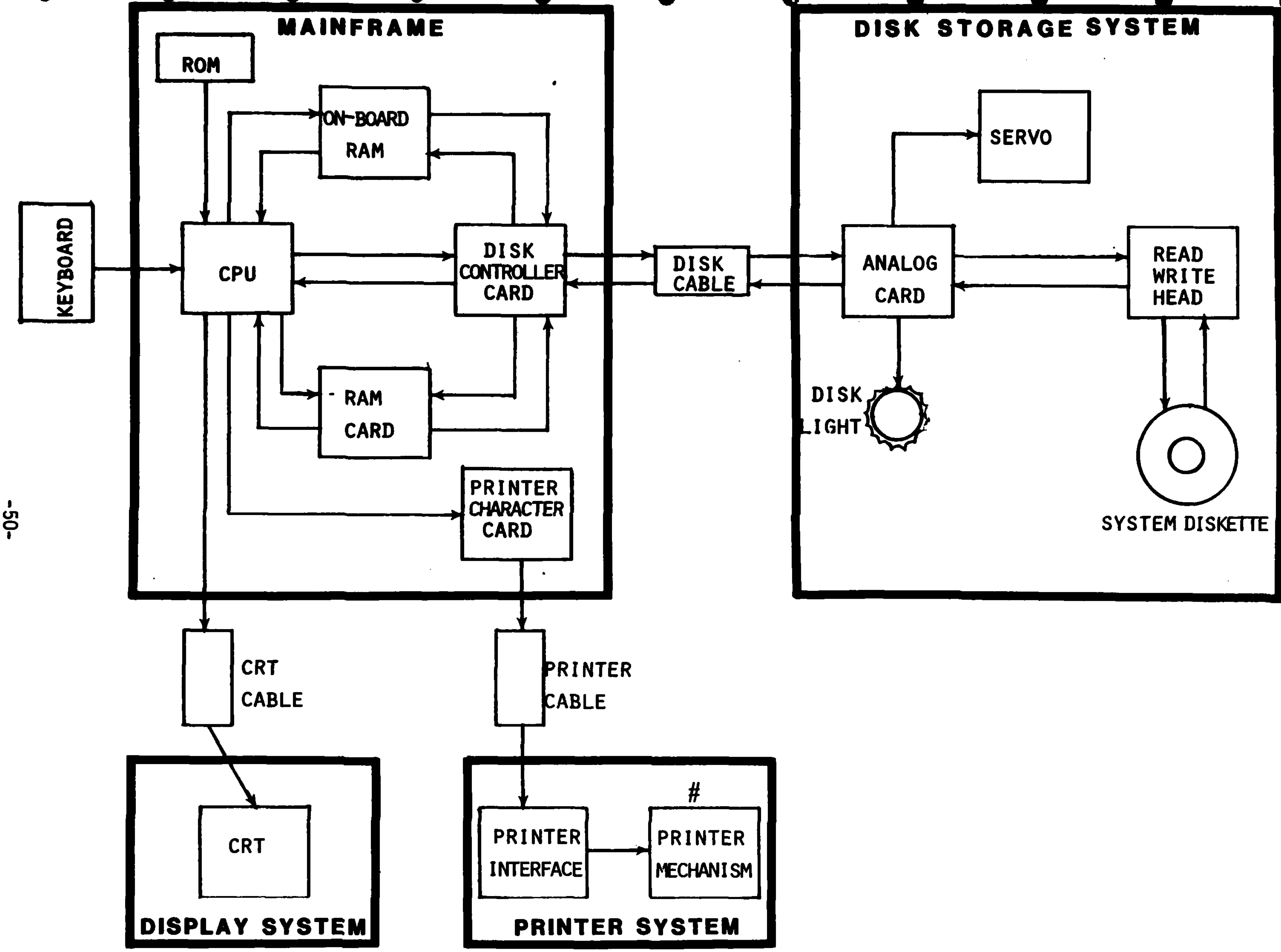
	Problem Sequence									
Subject	0	1	2	3	4	5	6	7	8	9
1	0A	7B	8B	3B	1A	4B	6A	5A	2A	
2	0A	4A	1B	8A	7A	6B	2B	3A	5B	
3	0A	7B	8A	4B	2A	3A	6B	5A	2B	1A
4	0A	7A	8B	1A	4A	3B	6A	2B	5B	
5	0A	8B	7B	4A	1B	6A	3A	5B	2A	
6	0A	8A	7A	1B	4A	6B	3B	2A	5A	
7	0A	4A	1B	7A	8B	5B	2A	3A		
8	0A	4B	1A	8A	7A	5A	2B	6B	3B	
9	0A	1A	4B	7B	8A	2A	5B	3B	6A	
10	0A	1A	4A	8B	7B	2B	5A	6A	3A	

('A' is the enhanced design; 'B' is the restricted design)



Appendix D

BLOCK DIAGRAM SYSTEM, STUDY TWO



ALWAYS GOOD

Appendix E INFORMATION FLOW

OFFICE OF NAVAL RESEARCH

Engineering Psychology Group

TECHNICAL REPORTS DISTRIBUTION LIST

Captain Paul R. Chatelier
Office of the Deputy Under Secretary of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, D.C. 20301

Engineering Psychology Group
Office of Naval Research
Code 442 EP
Arlington, VA 22217 2

Communication & Computer Technology Programs
Code 240
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Manpower, Personnel & Training Programs
Code 270
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

CDR Paul Girard
Code 250
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Information Sciences Division
Code 433
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Special Assistant for Marine Corps Matters
Code 100
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Dr. J. Lester
ONR Detachment
495 Summer Street
Boston, MA 02210

CDR James Offutt, Officer in Charge
ONR Detachment
1030 East Green Street
Pasadena, CA 91106

Director Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D. C. 20375

Mr. Philip Andrews
Naval Sea Systems Command
NAVSEA 61R2
Washington, D. C. 20362

Larry Olmstead
Naval Surface Weapons Center
NSWC/DL
Code N-32
Dahlgren, VA 22448

Mr. Milon Essoglou
Naval Facilities Engineering Command
R&D Plans and Programs
Code 03T
Hoffman Building II
Alexandria, VA 22332

CAPT Robert Biersner
Naval Medical R&D Command
Code 44
Naval Medical Center
Bethesda, MD 20014

Dr. George Moeller
Human Factors Engineering Branch
Submarine Medical Research Lab
Naval Submarine Base
Groton, CT 06340

Dr. Richard Cantone
Naval Research Laboratory
Artificial Intelligence Center
Code 7510
Washington, D. C. 20375

Dr. Robert Blanchard
Navy Personnel Research and
Development Center
Command and Support Systems
San Diego, CA 92152

CDR J. Funaro
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Mr. Stephen Merriman
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Mr. Jeffrey Grossman
Human Factors Branch
Code 3152
Naval Weapons Center
China Lake, CA 93555

Aircrew Systems Branch
Systems Engineering Test
Directorate
U. S. Naval Test Center
Patuxent River, MD 20670

Mr. Harry Crisp
Code N 51
Combat Systems Department
Naval Surface Weapons Center
Dahlgren, VA 22448

CDR C. Hutchins
Code 55
Naval Postgraduate School
Monterey, CA 93940

Office of the Chief of Naval
Operations (OP-115)
Washington, D. C. 20350

Dr. Julie Hopson
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Dr. Robert G. Smith
Office of the Chief of Naval
Operations (OP987H)
Personnel Logistics Plans
Washington, D. C. 20350

Mr. Norm Back
Combat Control Systems Department
Code 35 3512
Naval Underwater Systems Center
Newport, RI 02840

Human Factors Department
Code N-71
Naval Training Equipment Center
Orlando, FL 32813

CDR Norman E. Lane
Code N7A
Naval Training Equipment Center
Orlando, FL 32813

Dr. Robert Fleming
Human Factors Engineering Branch
Code 8231
Naval Ocean Systems Center
San Diego, CA 92152

Dr. Gary Pook
Operations Research Department
Naval Postgraduate School
Monterey, CA 93940

M. H. Taklington
Ocean Engineering Department
Naval Ocean Systems Center
San Diego, CA 92152

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D. C. 20380

LT Dennis McBride
Aerospace Psychology Department
Naval Aerospace Medical Research Lab
Pensacola, FL 32508

Human Factors Technology Administrator
Office of Naval Technology
Code MAT 0722
800 North Quincy Street
Arlington, VA 22217

Commander Naval Air Systems Command
Human Factors Programs
NAVAIR 334A
Washington, D. C. 20361

Commander Naval Air Systems Command
Crew Station Design
NAVAIR 5313
Washington, D. C. 20361

Dr. Edgar M. Johnson
Technical Director
U. S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Technical Director
U. S. Army Human Engineering Labs
Aberdeen Proving Ground, MD 21005

U. S. Air Force Office of
Scientific Research
Life Sciences Directorate, NL
Bolling Air Force Base
Patterson AFB, OH 45433

Chief, Systems Engineering Branch
Human Engineering Division
USAF AMRL/HES
Wright-Patterson, AFB, OH 45433

Director, Human Factors Wing
Defense and Civil Institute of
Environmental Medicine
P. O. Box 2000
Downsview, Ontario M3M 3B9
CANADA

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 12

Dr. M. Montemerlo
Human Factors & Simulation
Technology, RTE-6
NASA HQS
Washington, D. C. 20546

Dr. Jesse Orlansky
Institute for Defense Analyses
1801 North Beauregard Street
Alexandria, VA 22311

Dr. T. B. Sheridan
Deparment of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Dr. Arthur I. Siegel
Applied Psychological Services, Inc.
404 East Lancaster Street
Wayne, PA 19087

Dr. Harry Snyder
Department of Industrial Engineering
Virginia Polytechnic Institute and
State University
Blacksburg, VA 24061

Dr. Robert T. Hennessy
NAS - National Research Council
(COHF)
2101 Constitution Avenue, N. W.
Washington, D. C. 20418

Dr. Azad Madni
Perceptronics, Inc.
6271 Variel Avenue
Woodland Hills, CA 91364

Dr. James H. Howard, Jr.
Department of Psychology
Catholic Universtiy
Washington, D. C. 20064

Dr. Christopher Wickens
Department of Psychology
University of Illinois
Urbana, IL 61801

Dr. Edward R. Jones
Chief, Human Factors Engineering
McDonnell-Douglas Astronautics Co.
St. Louis Division
Box 516
St. Louis, MO 63166

Dr. Babur M. Pulat
Department of Industrial Engineering
North Carolina A&T State University
Greensboro, NC 27411

Dr. Stanley N. Roscoe
New Mexico State University
Box 5095
Las Cruces, NM 88003

Mr. Joseph G. Wohl
Alphatech, Inc.
3 New England Executive Park
Burlington, MA 01803

Dr. William B. Rouse
School of Industrial and Systems
Engineering
Georgia Insititue of Technology
Atlanta, GA 30332

Dr. Richard Pew
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02238

Dr. Douglas M. Towne
University of Southern California
Behavioral Technology Laboratories
1845 South Elena Avenue
Redondo Beach, CA 90277

Dr. Kay Inaba
XYZYX Information Corporation
21116 Van Owen Blvd.
Canoga Park, CA 91303

FIL