IMPLEMENTATION OF INSTRUCTIONAL FEATURES ON MAINTENANCE TRAINERS

Larry Rude
Chief Engineer
Honeywell Inc.
West Covina, California

SMSgt. William Lippke
Research and Development Engineering Technician
F-16 System Program Office
Wright Patterson Air Force Base

ABSTRACT

Automated instruction aids on a training device can significantly enhance the effectiveness of the device. However, the requirements for these aids (Instructional Features) must be consistent with the intended use of the trainer. If they are not consistent, the system implemented may be either more complex than required, or totally inadequate. This paper describes the development of Instructional Features where this inconsistency did exist. In this case the general processing requirements for student monitoring, student feedback, instructor reports and instructor controls were established. However, the specific in-classroom use of each was not. When the specific requirements were established, they were significantly less than the general processing requirements implied. The system design did meet both the general and specific requirements. However, a simpler approach would have satisfied the actually-used Instructional Features. This clearly shows the need to consider the specific classroom use of Instructional Features not just the general processing requirements.

INTRODUCTION

This paper presents the development of Instructional Features for a Simulated Aircraft Maintenance Trainer (SAMT). The initial concept, the software system design and implementation, an analysis of actual application versus the original intent and the factors to be considered in the application of Instructional Features to other trainer systems are discussed. The term Instructional Features for this paper is constrained to means the amount of automated monitoring of a student's progress through a task, feedback provided to the student, post lesson reports for the instructor, and instructor control of information presented to the student.

The trainer system for which the Instructional Features were developed is composed of ten different standalone trainers. The ten trainers correspond to the following aircraft subsystems: fire control, flight control instruments, navigation, electrical, environmental, hydraulics, weapon control, engine start, engine diagnostic and engine operating procedures. Each trainer consists of a master simulation control console (MSCC) and a simulation panel set (SPS). Figure 1 is the block diagram for a typical trainer. The MSCC contains the Honeywell Level 6 computer, a mass storage device, a lineprinter, a CRT and keyboard and a 35 MM projection system. The MSCC is identical for each trainer. The SPS is unique for each trainer and consists of one or two panels. On the SPS are the simulated aircraft controls and indicators and simulated test sets for the aircraft subsystem involved. Graphics on the panels provide for location of the controls and indicator on the aircraft. In addition to the aircraft and test set controls and indicators, there are action and element switches on the panels. The action switches provide the capability for the student to indicate his knowledge that a certain action, such as hose/cable connection, operation of hand pumps, aircraft safing actions, etc., is required as he proceeds through a task. The element switches provide to the student the capability to indicate which component he has isolated as being faulty or to call up a 35 MM slide of a particular component.

INITIAL INSTRUCTIONAL FEATURES CONCEPT

The initial objectives for the Instructional Features were to support: (1) student testing, (2) lecturing, (3) fault isolation treeing, and (4) self-directed learning. To accomplish these objectives, the Instructional Features were to provide: (1) monitoring of a student's progress through a tree structured sequence, (2) student feedback, (3) student testing, (4) instructor reports, and (5) instructor control over these features.

- The student monitoring was to be at a level that the trainer could uniquely respond to and/or record a student's selection of each branch and/or step in a Job Guide (JG) or Fault Isolation Manual (FIM).

- Feedback to the student was to be a CRT message, 35 MM slide, an audible sound, or any combination of these. The feedback was provided to cue the student, warn the student, or suspend execution of the lesson as a result of student branch selection or action taken on the trainer. The cues providing instructions and checkout aids required to lead the student through the training exercise (a set of procedures from the JG and/or FIM). The warnings being for procedural and operational errors, systems malfunctions and safety violations.

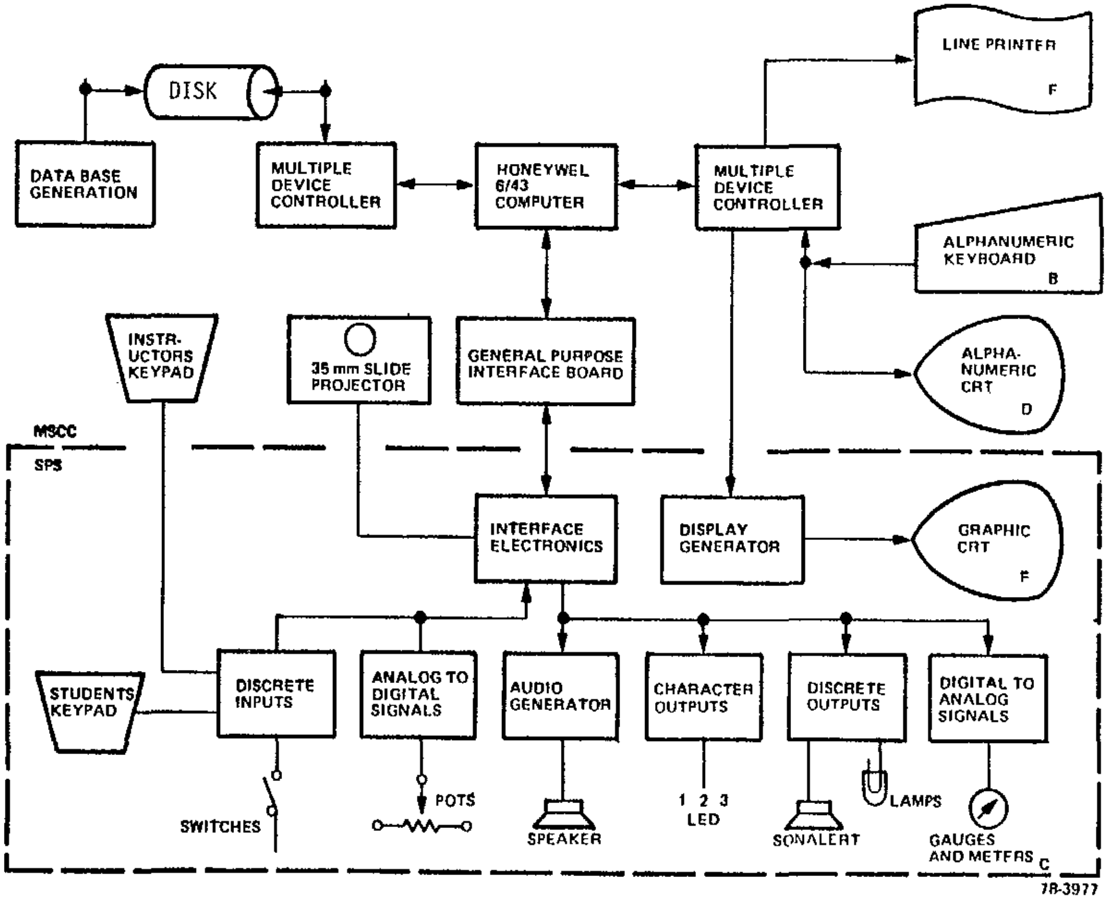

Figure 1. F-16 SAMT Interface Block Diagram

• Student testing was the presenting of questions with the follow-on sequence being determined by the student's response.

• The instructor reports were to consist of student responses to test questions, procedural and operational errors made, safety errors, actions taken on the trainer and branch selections made.

• Controls provided to the instructor were to consist of malfunction selection, entry and modification of plans of instructions, amount of feedback and modification of testing materials, self-directed learning materials (cues and warnings) and fault isolation schemes. Plans of instructions were the mix and ordering of student testing, lecturing, and fault isolation treeing. The amount of feedback to be controlled by levels of aiding selected by the instructor.

SOFTWARE SYSTEM DESIGN

The software system design and implementation to meet the Instructional Features requirements consisted of a Courseware Authoring Language Generator (CALGEN) and an on-line real time interpreter (Procedure Monitor). The actual monitoring, recording and feedback is specified in the CAL which is a high level source language. The source is input to the CALGEN, checked for errors and decoded into an object code. The interpreter decodes the CAL object code for each student action determining the correctness, providing the student feedback and recording the action.

The CAL was designed to provide a friendly interface to the training analyst developing the step-by-step training exercise for the student. This software (courseware) being based on the step-by-step actions called out in the aircraft technical orders and/or job guides.

The basic language constructs are:

STRUCTURAL COMMANDS

```
PROCEDURE <NAME>
GLOBAL VARIABLES ARE <LIST>
LOCAL VARIABLES ARE <LIST>
   MONITOR <NAME> WATCHING <ACTION LIST>
   END MONITOR <NAME>
END PROCEDURE
```

ACTION EVALUATION/RESPONSE COMMANDS

```
EXPECT <ACTION LIST>
ALLOWING <ACTION LIST>
UNLESS <ACTION LIST>
SET <EQUIPMENT NAME / VARIABLE> TO <STATE / VALUE>
```

ACTION EVALUATION/RESPONSE COMMANDS (CONT.)

```
PROJECT <N> FOR <HIGH / MED / LOW> AIDING
DISPLAY <MESSAGE>
AT <PRINTER / DISC / CRT>
FOR <HIGH / MED / LOW> AIDING
```

CONTROL COMMANDS

```
INVOKE <MONITOR NAME> AT PRIORITY <N>
FORCE <MONITOR NAME> TO <SUSPEND / RESUME>
TERMINATE <MONITOR NAME>
IF <CONDITION> THEN
GO TO <LABEL>
ESCAPE <N>
```

WRITER'S CRAMP COMMANDS

```
ABBREVIATE <STRING> MEANS <STRING>
INCLUDE <FILE NAME>
MAP SLIDE <PHYSICAL NUMBER = LOGICAL NUMBER LIST>
```

These constructs can be used to provide student monitoring with error detection, aiding and equipment simulation. An example of this is:

```
L:  EXPECT SAFETY_LATCH

EXPECT SAFETY_LATCH = ON  ALLOWING NONE
UNLESS NONE

   IF CORRECT THEN GO TO L:SAFE_TO_PROCEED
   END IF

   (ELSE THE WRONG ACTION WAS TAKEN)

   DISPLAY "Place safety latch on" AT CRT
   FOR LOW AIDING

   GO TO L:EXPECT_SAFETY_LATCH

END EXPECT

L:  SAFE_TO_PROCEED

     SET READY_TO_PROCEED_LIGHT TO ON
```

```
L:  SAFE_TO_PROCEED (CONT.)

     PROJECT 23 FOR HIGH AIDING

              .

              .

              .
```

CAL can be used to provide student testing. An example of this is:

```
L:  DISPLAY_COOLING_AIR_QUESTION

    DISPLAY "Should cooling air be applied
            before the main electrical
            power supply is hooked up?" AT
            CRT

    EXPECT C_IRN = AN_KYBD_STRING ALLOWING

    ANY UNLESS NONE END EXPECT


    IF AN_KYBD_STRING = 'NO' THEN

       DISPLAY "Error in choice of cooling
               air application" AT DISK

       DISPLAY "T.O. JG-11-00 provides in-
               formation concerning the
               application of cooling air.
               Would it be proper to recon-
               sider your last answer?"  AT
               CRT


    EXPECT AN_KYBD_STRING = 'YES' GO
    TO L:DISPLAY_MENU END IF

    (A more specific aid is needed)

    DISPLAY "On page 59 of T.O. JG-11-00
    there is information concerning the
    application of cooling air" AT CRT

    GO TO L:DISPLAY_MENU

    END IF

L:FRONT_PANEL_QUESTION

    DISPLAY "Should the front panel be
            removed before the application
            of the main electrical power
            supply?"

              .

              .

              .
```

As can be seen from the language constructs and by these examples, the courseware can be constructed to provide monitoring of student performance by comparing actual actions against a predefined:

- Single action
- Set of actions which must be performed sequentially
- Set of actions which may be performed in any order
- Set of actions which are extraneous and to be ignored
- Number of actions
- Time

It provides presentation of:

- Simulated device responses
- Pictorial material
- CRT messages

CAL provides feedback for:

- Correct actions
- Incorrect actions
- Time
- Number of actions

Figure 2 describes the courseware generation process and the interface to the on-line trainer software. The first step in this process is the annotation of the T.O.s or Job Guides. Annotation is the specifying of the student monitoring, feedback responses, error processing and recording requirements for each step in the T.O.s. The CAL source is coded based on the annotated T.O.s. CALGEN is used to compile the source code. The object code generated is produced as a set of files on the disk used by the operational software. The object code is interpreted by the on-line, real time trainer software providing the desired student monitoring, feedback and report generation.

The on-line SAMT software system provides two major functions: a simulation model function and a trainer control function. The simulation model simulates normal and malfunctioning aircraft system operation by supplying appropriate responses to simulation panel set (SPS) inputs. The trainer control function has overall control of the training function including initialization, training exercise preparation, training exercise presentation, monitoring of the student action and comparison of the action for compliance with the courseware.

In addition, input and output subfunctions are provided which condition and control the data flow to and from the computer. A Powerfail function provides for resumption of a problem exercise interrupted by a power failure. An operating system provides a multi-tasking environment, standard peripheral interfaces and file management. A Kernel component which provides interfaces between the other software components and the operating system. The Kernel dispatches execution according to priority, state of the training system and real time.

Figure 2

COURSEWARE PREPARATION FLOW

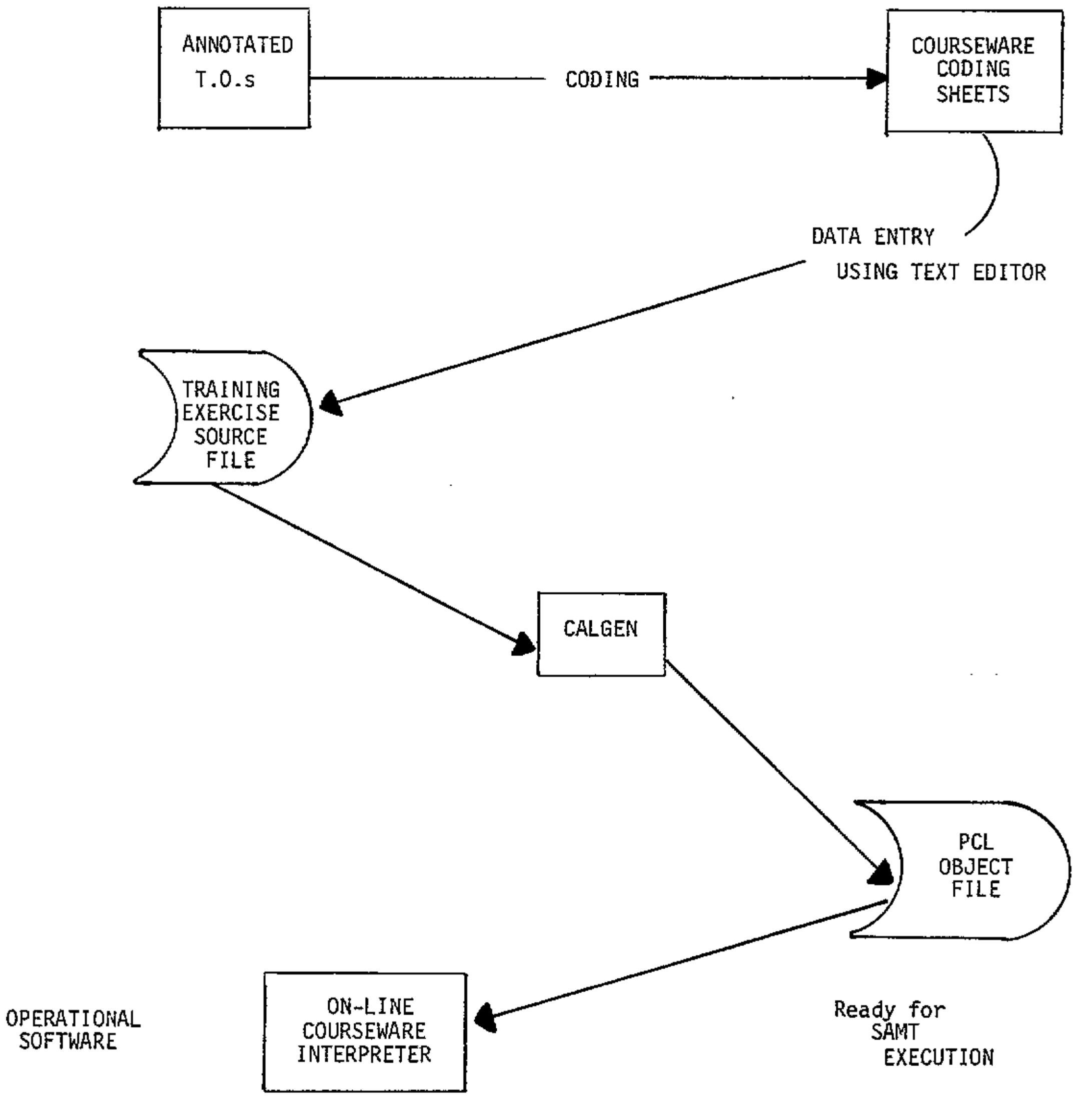

Figure 3 shows the operational software hierarchy. The Instructional Features requirements are satisfied within the trainer control function. The control courseware provides services generic to the SAMT. Training courseware provides T.O. unique information. The Procedure Monitor is the on-line interpreter of the courseware. The training courseware specifies what student inputs (from the panel or the keyboard) and in what order these actions are to occur. The training courseware can also specify what response is to be generated for a given action. The response may be a display on the SPS, a CRT message, 35 MM slide and/or problem freeze. Thus providing equipment simulation, student aiding, error feedback and/or student testing. Training courseware also specifies what information is to be recorded in the student record.

The control courseware controls the training courseware and provides the instructor control. These controls are:

1. problem exercise list
2. message modification
3. status report call up
4. freeze enable

The problem exercise consists of up to 15 elements which define an exercise to be performed by the student. These elements are:

1. procedure -- Job Guide or fault isolation procedure contained in the training courseware.

2. malfunction

3. aiding level -- The student aiding level may be low, medium or high.

4. parameter -- simulation variables such as temperature, oil pressure and fuel quantity.

5. Sign IN -- Instruction to the student to enter his name and identification. This information is then entered in the training record.

6. time limit -- Alloted time for the student to complete the exercise. If the student exceeds this time, a message is displayed and the exercise halted.

7. action limit -- number of action steps to be taken by the student in performing the exercise. If this number is exceeded, a message is displayed and the exercise halted.

8. lesson -- precanned problem exercises stored on the disk.

These elements are entered by the instructor in the order he desires. The list is then processed sequentially. This allows the instructor to select a combination of malfunctions and procedures to be performed and to specify under what conditions (aiding level, parameters, time and number of actions). The lesson element provides for routinely performed training sessions without the instructor having to reenter all elements each time. A typical problem exercise might be:

1. Sign IN
2. aiding level high
3. temperature parameter 73°
4. action count 75
5. malfunction 1
6. procedure XXX
7. procedure YYY
8. malfunction 3
9. procedure AAA
10. procedure BBB

Items 6 and 9 being the operational check procedure which would detect the malfunction and items 7 and 10 being the fault isolation procedures for the specific malfunction.

The message modification control allows the instructor to temporarily modify messages contained in the training courseware.

The status control allows the instructor to request display of the training record. The training record consists of the date and the sequentially logged data consisting of P.E. list item, state of system freeze, malfunction, aiding level, actual time, action counts, student name, and identification and items directed to be recorded by the training courseware.

Freeze control allows the instructor to enable/disable halting of the exercise automatically when the student commits a hazardous error.

SPECIFIC REQUIREMENTS VS. CONCEPT

The software system was designed based on the initial Instructional Features requirements. When the detailed Instructional Features objectives (specific use of the trainer) were available, it was obvious that they were significantly less than the original concept. The specific objectives of the student monitoring requirements consisted of:

1. critical/hazardous actions -- Monitor the warnings and cautions in the Job Guides.

2. completion criteria -- Monitor that a few specific actions were accomplished by the student before allowing him to complete a procedure. The specific actions may be procedure unique.

3. malfunction removal prerequisites -- these prerequisites are not procedure dependent but are airplane subsystem dependent. It consists of monitoring for certain actions being accomplished before the student is allowed to identify which component he believes has failed.

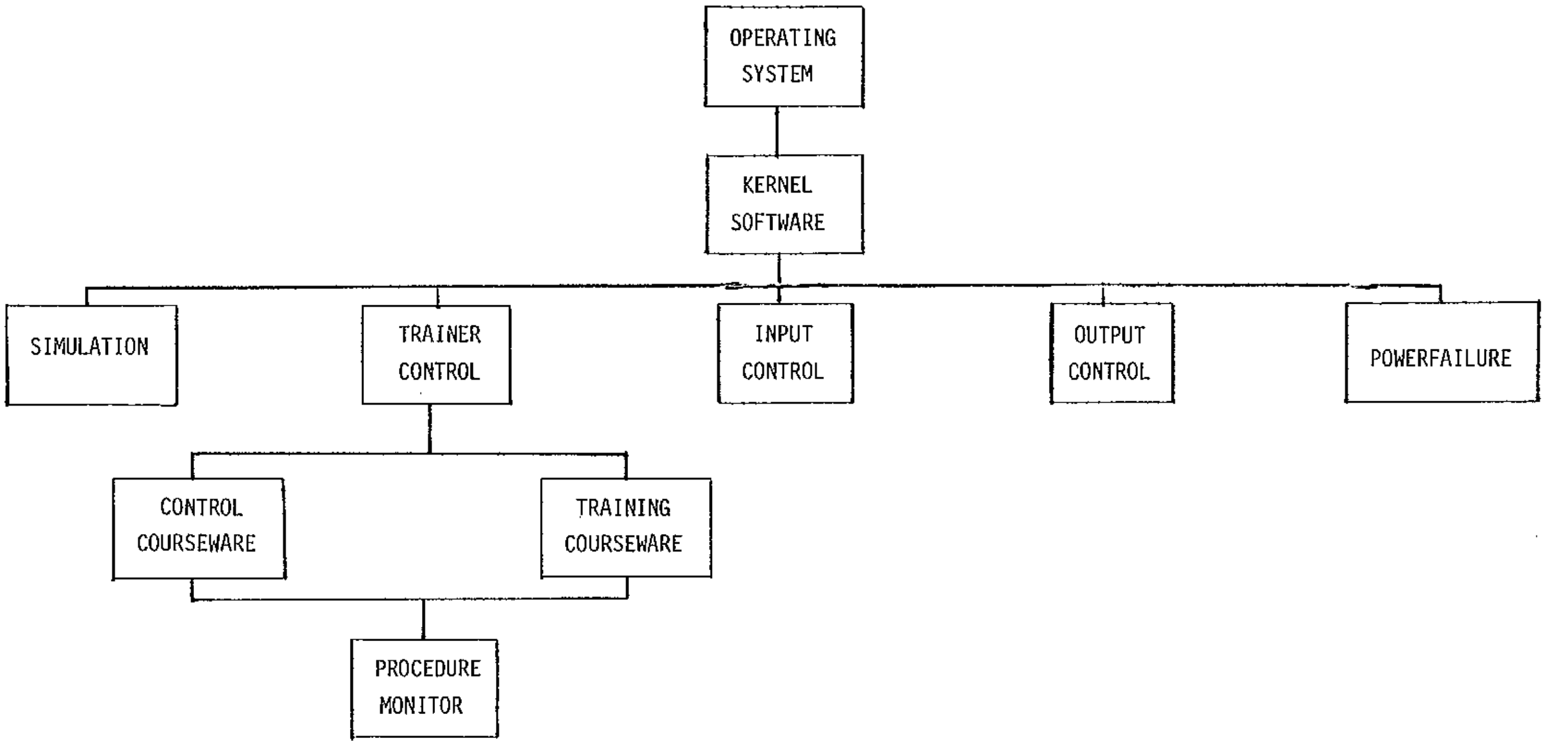

Figure 3. Software Hierarchy

4. time monitor -- Monitor the length of time the student required to complete a procedure.

5. action monitor -- Monitor the number of actions the student made in completing a procedure.

The reasons for the significant difference between the initial concept of the Instructional Features and the final concept fall into three areas:

1. The general requirements for the training had been determined, but the specific in-classroom use of the trainer had not been established. The specific uses were not determined until the subject matter expert (SME) was asked to define the intended use of each Instructional Feature. This was done to allow the specific training courseware for each procedure to be specified and generated.

2. The full capability of the Instructional Features was not required by the SMEs for two reasons:

a) The instructors are present during the training exercise and can provide the aiding necessary.

b) The Job Guides have a high rate of change for a new weapon system, therefore making it difficult to keep step-by-step courseware monitoring current with the Job Guide.

3. The cueing or student aiding requirement was more state dependent than Job Guide step dependent than was originally thought. That is, a cue is required at the completion of a task or subtask at each step.

APPLICATION OF INSTRUCTIONAL FEATURES

In some cases, the development of a system to meet a larger spectrum of Instructional Features may be desirable. In other cases this may not be true. There are several factors which must be considered in making that decision. Some of these factors are:

1. Is the system for which the maintenance trainer is being developed mature? In this case, the T.O.s are available and have a low rate of change.

2. Is the training to be performed for theory of general system operation? In this case, representative T.O.s or procedures can be used and the changes of the actual equipment T.O.s need to be incorporated only when they represent changes in the theory of operation.

3. Is the training objective, familiarization with the T.O. system? Again, representative T.O.s can be used.

4. What is the level of training to be provided? Entry level or basic training requires more aides and cues to assist the student initially and then gradually remove the crutches.

5. Is it desired to have uniformity of training, that is, less dependent on individual instructors? The trainer providing at least the minimum aiding and evaluation ensures all students receive minimum level of instruction.

6. Is there a high student to instructor ratio? In this case, both the student and the instructor need help (automatic student aiding and monitoring and student performance reports).

7. Is the requirement conversion training, that is, training a B-52 maintenance man to maintain B-1's? In this case, he is familiar with the T.O. system. He will group tasks rather than performing them in series. The student does this because he is able to look ahead. For example, he knows that he can make all connections to a given piece of equipment at once and save time rather than when they are called out in the T.O. In this case, procedural aiding does not apply.

8. Is it desired to have consistent training at more than one level? In this case, where training is to be provided for entry as well as "conversion training," the Instructional Features need to be and can be diminishable and tailorable to meet all levels.

SUMMARY

The system developed to meet the initial concept of Instructional Features does meet those requirements as well as the specific final requirements. This fact shows that the Instructional Features for a trainer system can be designed to meet several applications. However, it is obvious that a less sophisticated system would have met the final requirements. As can be seen from this example, Instructional Features must be determined in two phases: 1) determine the general features required by evaluating the intended use of the trainer, and 2) evaluate the specific classroom use of each feature.

ABOUT THE AUTHORS

Mr. Larry Rude is a Chief Engineer with Honeywell Inc.'s Training and Control Systems Operation (T&CSO). He is currently responsible for training systems software development. He has a Bachelor's Degree in Applied Mathematics from the University of Idaho. He was formerly the Maintenance Trainer Software Section Head for Honeywell, where he was responsible for the software development on the F-16, F-15 and AWACS-Radar Maintenance Trainers. Prior to joining T&CSO, he was a Staff Engineer for Honeywell's Avionics Division. Before coming to Honeywell, Mr. Rude was a Captain in the Air Force involved in the development of real time software for checkout and launch of Air Force Satellites.

SMSgt. William Lippke is a Research and Development Engineering Technician in the F-16 System Program Office at Wright Patterson Air Force Base where he is responsible for the engineering management of all Maintenance Training devices for the Multi-National F-16 fighter aircraft. He has 21 years of active duty in the

Air Force; the last 11 have been in training equipment development and acquisition. He holds an Associate Arts Degree in Electronic Engineering Technology. He is a past Chairperson of the Maintenance Training session of the Interservice/ Industry Training Equipment Conference.