N96-12949

536-61

p. 12

UFC Advisor: An AI-Based System for the Automatic Test Environment

David T. Lincoln *
Pamela K. Fink, Ph.D.

Southwest Research Institute
San Antonio, Texas 78228-0510
(512) 522-3368

Abstract

The Air Logistics Command within the Air Force is responsible for maintaining a wide variety of aircraft fleets and weapon systems. To maintain these fleets and systems requires specialized test equipment that provides data concerning the behavior of a particular device. The test equipment is used to "poke and prod" the device to determine its functionality. The data represent voltages, pressures, torques, temperatures, etc. and are called testpoints. These testpoints can be defined numerically as being in or out of limits/tolerance. Some test equipment is termed "automatic" because it is computer-controlled. Due to the fact that effective maintenance in the test arena requires a significant amount of expertise, it is an ideal area for the application of knowledge-based system technology. Such a system would take testpoint data, identify values out-of-limits, and determine potential underlying problems based on what is out-of-limits and how far. This paper discusses the application of this technology to a device called the Unified Fuel Control which is maintained in this manner.

* formerly with SAALC/MAT, Kelly A.F.B., San Antonio, Texas 78241-5000

Introduction

The Air Force maintenance capability is primarily organic in that Air Force personnel perform the diagnosis and repair tasks. Much of the test equipment and the devices they support were developed and fielded in the early- to mid-seventies. Thus, most of the equipment tends to be out-moded and no longer supported by the vendor. Therefore, use of such equipment to diagnose a device requires a certain level of expertise obtained over years of experience. For example, a minimum of ten years of experience is needed to produce an experienced diagnostician for the Unified Fuel Control (UFC).

The UFC is the "carburetor" for the F-100 engine, the engine that flies the F-15 and F-16 fighter jets. It is essentially a large, complex mechanical computer. Nearly 95% of all UFC's in the Air Force's inventory are repaired and tested at the San Antonio Air Logistics Center (SAALC) at Kelly A.F.B. The controls arrive at SAALC for one of two reasons: scheduled overhaul or unscheduled maintenance. A UFC will be scheduled for overhaul when it exceeds the Air Force's recommended maximum operating hours (MOH). Depending on whether the UFC is taken from an F-15, which has two engines, or an F-16, which has only one engine, and the configuration of the UFC, this MOH can vary from 1500 to 4000 hours. UFC's arrive for unscheduled maintenance due to a malfunction that can be caused by a variety of problems. When a UFC arrives from the field it has a processing tag attached to it. This tag contains the problem description as reported by the field, which ranges from very specific (e.g. broken lever arm) to very vague (e.g. does not work).

Determining what could be causing a malfunction can be very difficult. The UFC is composed of over 4500 parts, many of which can cause the control to fail. The test equipment used to maintain the UFC is a customized piece of automatic test equipment and is referred to as a test stand. A test stand is analogous to an electronic diagnostic system one might find at a car repair shop. The UFC is connected to the test stand and run through a series of tests to determine its weaknesses, just as a car's engine might be. An expert in diagnosing the UFC must take into account not only potential problems with the UFC, but the possibility that the test stand may not be within calibration standards. In addition, the UFC is maintained by a set of four different test stands, each with a specific set of test procedures to help diagnose certain parts of the UFC. Thus, the number of possible failures and their underlying symptoms is large, creating a need for very domain-specific expertise.

The UFC Maintenance Process

To standardize the decision making strategy for the maintenance process of the UFC, SAALC uses the concept of On-Condition Maintenance (OCM). This concept is one in which a team of domain experts is chosen to make all decisions concerning the repair of a UFC as it passes through the maintenance process. These decisions are based on the UFC's condition upon receipt at the maintenance facility and at various points during testing. An overview of the entire maintenance process is given in Figure 1. There are six potential areas where knowledge-based system technology could be applied. They include the pre-RAR decision, the post-RAR decision, the Augmentor Body, Gas Generator, and Distribution Body decisions, and the post-M&I decision. Each of these systems would utilize the information available at a given point in the process to form recommendations about what should be done next.

The UFC maintenance process begins with a visual and electrical inspection. The results of these inspections, along with the field reported problem description, give the OCM team personnel a foundation for their first decision: overhaul, demate and repair, or run the Run-As-Received (RAR) test. To overhaul a UFC requires breaking the control down to its lowest levels and replacing defective parts as it is rebuilt. The average length of time required to do this is 650 hours. To demate and repair means to break down the UFC to one of its three major sub-assemblies (Augmentor Body, Gas Generator, and Distribution Body) and perform the prescribed repair actions.

The RAR test is actually a series of automatic tests that are run to give diagnostic information about what might be wrong with the UFC. It is hosted on a Data General computer and is run "hand's off" (i.e. no adjustments made as the test runs). The time required for this test averages seven hours but can go as long as twelve or fourteen. The computer, in turn, drives the test stand that "pokes and prods" the UFC. The RAR generates approximately 450 testpoints and records the UFC's value at each testpoint. The result of the RAR is a one inch thick document with the various testpoints grouped into related paragraphs which represent the three distinct sub-assemblies of the UFC. The RAR is then analyzed by one or more members of the OCM team and, based on this analysis and the team members' experience level, a recommendation is made as to the best repair action. This recommendation may include overhaul, demate and repair, or run the Mating & Indexing test (M&I). The M&I involves the calibration and adjustment of the UFC. If the UFC has been overhauled or demated and repaired, it is then reassembled and run through the M&I. The M&I and the RAR both test the UFC with the same tolerances. Once the M&I has finished, another iteration of decision making is made: overhaul, demate and repair, or run the Service Acceptance Test (SAT). The SAT is essentially the same test as the RAR and M&I with a different set of tolerances. Once the UFC passes the SAT, it is returned to the Air Force inventory.

Although three shifts are required to meet the demand for UFC production, the OCM team is only available during the first shift. During the second and third shift and on weekends, test recommendations are left up to the line or shift supervisors, or the UFC is put on hold until an OCM team member is available. Thus, delays are inevitable in obtaining a diagnosis for a UFC. A crucial task performed by the OCM team that is vital to an accurate diagnosis is visually identifying all testpoints on the RAR that are out of limits. Due to the stress that is placed on the OCM team to produce, there is a good probability that some of the testpoints that are out of limits are not identified. This naturally leads to erroneous and inconsistent decisions.

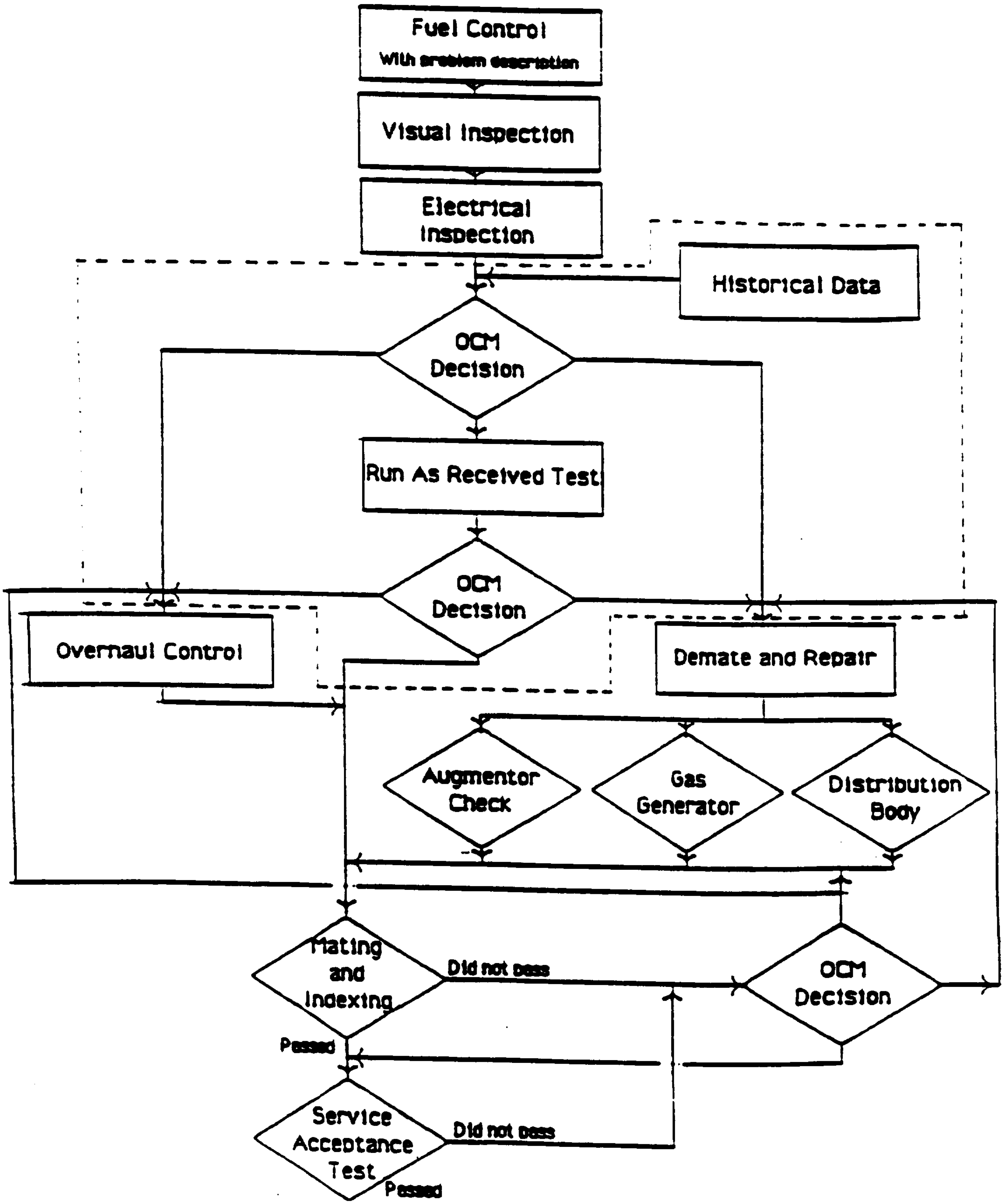

FIGURE 1. Overview of the UFC maintenance process.

Issues Concerning the Development a Knowledge-Based System for the Automatic Test Environment

Of the six potential areas where a knowledge-based system could be implemented, the pre-RAR and post-RAR (hereafter referred to as the UFC Advisor) were selected to start with because they are procedures that most UFC's must undergo and because the problems of integration into an existing test environment were not so severe. These initial phases of the maintenance process are not highly interactive and so did not have to be performed out on the shop floor next to the test stand (a volatile environment). The pre-RAR system is basically a front-end to the historical database shown in Figure 1 that allows the user to enter preliminary data about each UFC as it comes in from the field and to obtain the data on the UFC from previous repair actions.

The UFC Advisor was developed as an effort to streamline the maintenance process and increase the production of UFC's at Kelly A.F.B. Since the experts perform diagnoses from a problem-oriented standpoint, the UFC Advisor is designed to mimic this approach. It makes recommendations based on the RAR test results and furnishes three benefits with respect to the RAR:

- ensures identification of all testpoints out of tolerance
- provides consistent recommendations
- reduces time lost due to the unavailability of the OCM team on second and third shifts

The UFC Advisor was developed as a joint effort between civil service computer scientists and engineers and researchers from Southwest Research Institute. This cooperative effort was one in which the civil service employees acted as apprentices to the more experienced researchers, with the intention that the Air Force would gain an organic capability in artificial intelligence/knowledge-based systems development.

As with any knowledge-based system development, a decision had to be made as to the type of hardware that would host the system and, since many knowledge-based system shells/languages are hardware dependent, which shell or language would best fit the needs for the UFC Advisor. Additionally, data acquisition from the UFC test stands was non-trivial. As stated before, much of the test equipment used in the maintenance process in the Air Force is out-dated. This is true of the UFC test stands. Because these stands are so old, the test data generated is often only accessible at the test stand. This is not a problem when a human is interpreting the test data since he/she can easily read the test stand's screen or the printout to obtain the testpoint out-of-limits data. However, acquisition of such data electronically could be very difficult.

The ideal solution would have been to host the UFC Advisor on the Data General computers that run the test stands, but these computers, which were designed and implemented in the mid-seventies, have only 256k of RAM with memory virtually exhausted and no capacity for expansion. The development team also concluded that the UFC Advisor would be too large to run in a PC environment and so decided that a workstation would be suitable since a workstation has both the memory and speed required to run a system as large as the UFC Advisor. In addition, a workstation is less expensive and more compact than a mainframe. After comparing the Apollo, SUN, and VAX workstations, the SUN was chosen for development. Due to an unexpected hindrance, the development team realized that it would take six months for SUN to deliver the workstations. Thus, an interim decision was made to prototype what would fit of the UFC Advisor on an IBM PC. Then, upon arrival of the workstations, the knowledge could be transferred from the PC to the SUN and expanded to completion.

As to the choice of a software language tool, CLIPS was chosen over many others for a variety of reasons. First, CLIPS was available so it was chosen as the tool to use for development of the initial prototype on the PC. The development team also knew of CLIPS' portability and decided to continue to use it since there was no reason to believe that CLIPS code designed on the PC would not run on the SUN. Second, acquisition of software by the government is slow. In view of the fact that CLIPS is supplied to government agencies at no cost, the normal delay expected to obtain a specialized knowledge-based system development tool such as CLIPS is eliminated. Another advantage CLIPS possesses is its capability of being embedded in an application program written in a conventional language such as C.

Once CLIPS was chosen the next step was to acquire the data from the test stands. As stated before, this acquisition turned out to be very difficult. The initial suggestion was to take the RAR data from the Data General and port it to the SUN, but again the Data General's are virtually out of memory and thus had no capacity to host another software progam which might write the RAR data into a format understandable to the SUN. The next idea was to eavesdrop on each test stand's printer and capture the RAR data with a PC located at each test stand as the data printed out to the printer and then transfer the data by floppy to the SUN. But the Air Force's requirement that any computer equipment located in the test stand area be enclosed in plastic because of the explosive nature of the fuel used to test the UFC, along with the fact that there are over twenty UFC test stands, made it economically unreasonable to use this approach. It was also unrealistic to expect an OCM team member to type in over 450 testpoint values at a terminal. It was still necessary, though, to acquire the data quickly since the RAR data remains memory resident for only thirty minutes. Given shift changes, employee's lunch and scheduled breaks and other unforeseen delays, many of the RAR's could be lost.

The solution decided upon was to monitor each test stand's printer through a series of specialized buffering hardware. The data is shipped over an ethernet that connects each test stand to one of several communications boxes. These boxes then ship the data to a single PC where the data is identified by UFC serial number and undergoes preliminary analysis, storing only what is needed. When it has been determined that all data for an RAR on a given UFC has been obtained, the file is closed and sent to the SUN where the UFC Advisor resides.

The UFC Advisor

The UFC Advisor essentially has no user interface. Under normal operations the system automatically receives over the network a file containing testpoint values from an RAR. When analysis is complete, the system prints out its final report. In case something does go wrong, however, the system does provide a facility for querying about the status of the data on all of the UFC's in the system at that point in time.

The UFC Advisor is a single executible program composed of three parts: a C program to preprocess the data input from the PC, a second C program to read the processed file and test all of the RAR testpoints for in- or out-of-limits condition, and a "diagnostic inference engine". Each of these programs will be discussed in detail. An overview of the total UFC Advisor system architecture in shown in Figure 2.

The preprocessor is essentially a parser and is designed to strip all irrelevant information from the file received from the PC. It also removes duplicate paragraphs, as an RAR may run the same paragraph more than once. If the file contains errors, it is copied into a directory to be corrected by an OCM team member. If there are no errors, the file is read by the second C program.

This second program begins by initializing CLIPS. Then it reads in each testpoint value and determines whether the testpoint is low, high or within limits based on a predefined minimum/maximum file. If the value is out-of-limits, then a string, which contains information such as which subsection (or paragraph) of the UFC contains the testpoint, the testpoint number, its out-of-limits value (i.e. high or low) and its actual value, is written into a "symptoms" files. Also, all testpoints, along with their recorded, minimum, and maximum values are written to an output file, with testpoints that are out-of-limits highlighted by an asterisk. This process is reiterated for every testpoint in the RAR. Upon completion, the diagnostic inference engine assumes control.

The diagnostic inference engine, which was designed and implemented in CLIPS, (ver. 4.2), is a seventy rule knowledge-based system. Each iteration of the system performs a series of steps. It has been designed as a generic diagnostic inference engine to handle association of testpoints out-of-limits with problems and solutions. First, it reads the "symptoms" file and asserts each string (or symptom) as a fact. An example of a fact is

P 9003 tp 10 ITEM PFN-PFCB OOL high RCRD 57

where 'P 9003' indicates paragraph 9003, 'tp 10' is testpoint 10, 'ITEM PFN-PFCB' is a subcategory of the testpoint, 'OOL high' means out-of-limits high and 'RCRD 57' is the recorded value for the testpoint. The second step involves loading into memory the knowledge that has been acquired from the experts. The knowledge is grouped by paragraph number, where each paragraph is stored in a separate file. It is in the form of CLIPS facts. This set of files comprises the test-specific knowledge base. Thus, to modify the knowledge base simply requires modification of the file which contains the information about the paragraph in question.

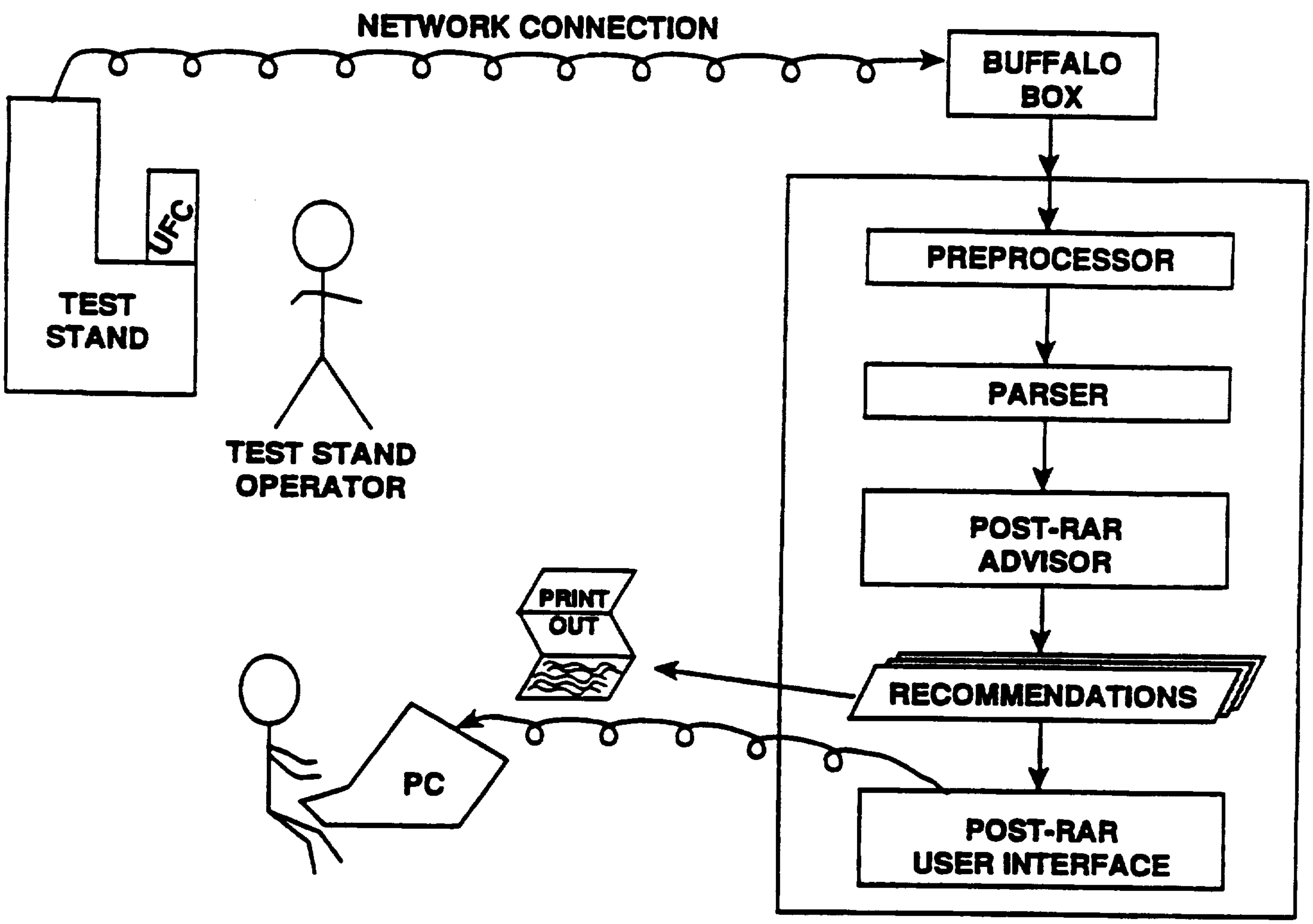

Figure 2. Configuration for the UFC-Advisor System

Since each paragraph is loaded as a fact, changes to the knowledge base do not require a recompilation of the rules. Each fact in the knowledge base has associated with it a symptom, the minimum and maximum value for the symptom's testpoint, a potential problem for that testpoint, evidence for that problem, a possible solution to the problem and the cost to perform that solution. For each symptom, there may be one or more symptom/problem/solution sets associated with it. An example of one of these facts is:

SYMPTOM: P 9003 tp 10 ITEM PFN-PFCB OOL high
MIN 37.5 RCRD dummy MAX 42.0
PROBLEM: Contamination of speed receiver orifice
EVID: 5
SOLUTON: Decontaminate speed receiver orifice
COST: 0.5

The third step of the diagnostic inference engine utilizes a set of rules that match each symptom from the first step with each symptom/problem/solution set in the second step. Each matching set is then retracted and reasserted with the RCRD field of 'dummy' replaced with the testpoint's actual value. Since many symptom/problem/solution sets have the same symptom associated with them, use of a value like 'dummy' prevents the system from only capturing the first occurrence of a matching set and bypassing the rest. Next, all unused symptom/problem/solution sets (i.e. those with 'RCRD dummy') are retracted to release memory. Many problems may have multiple symptoms and/or solutions and as mentioned before, the UFC Advisor attempts to diagnose from a problem-oriented standpoint.

To further complicate the diagnostic process, discussions with the experts revealed that key testpoints, when out-of-limits, forced repair actions that had to be dealt with immediately. This knowledge is referred to as meta-knowledge. A second set of testpoints, while not requiring immediate action, had priority over all others. Thus, a level of meta-knowledge, plus prioritization of the problems, became necessary. To handle the issues of meta-knowledge and prioritization, a method of evidence maintenance was used.

First, for each unique problem a tally is initialized. Then, all problems that match a tally are combined by combining their evidences. Also, if the paragraph affiliated with a specific symptom/problem/solution set is one with priority over the others, the evidence is multiplied by a "priority factor" before being added. After all sets have been tallied, they are sorted based on total evidence. Next, a set of "meta-rules" execute based on the meta-knowledge obtained from the experts. The purpose of firing these rules now and not initially is two-fold. First, the development team, following the expert's advice, decided to print out all recommendations rather than using a minimum threshold based on evidence. Second, by firing last the meta-rules can write directly to the output file as the first set of recommendations. Figure 3 gives an example of a portion of the UFC Advisor's output. A typical output is around ten to twelve pages.

Finally, the symptom/problem/solution sets associated with the tallies are written to the output file in order of evidence. As one can see from Figure 3, these sets may contain one or more solutions for each problem with one or more symptoms for each solution. Additionally, along with the minimum, recorded, and maximum values, the cost for each solution is written. Thus, the output consists of three parts: a summary of testpoint information, meta-rule recommendations, and all other recommendations, listed by priority.

Current Status

At the present time, all record keeping in the UFC maintenance area is paper-oriented. The current method for storing records is to package the RAR, M&I, SAT and all other written documentation into a plastic bag and store the package in a filing cabinet. Thus, to gather any statistical information such as a testpoint that is a recurring problem, occurrences of less frequent but highly critical repairs, or any correlation of testpoints out-of-limits to solutions is almost impossible.

The UFC Advisor as it currently stands, where it is capable of supporting the RAR test has the potential for saving considerable test stand and OCM team time each month. Based on an analysis of the entries in the UFC Test Log for the one month period of August 1989, 25% of the UFC's that came in had an RAR run, with an average run time of 18.2 hours. The average time spent after an RAR was run and waiting for a recommendation from the OCM team was 9.25 hours. The total wait time after an RAR was run was 360 hours, or approximately 15 24 hours days. This equates to half a test stand per month being wasted on just waiting on the decision that has to be made after an RAR is run. In addition, each RAR evaluation requires 30 - 60 minutes of an OCM team member's time. As a result, approximately 36 hours per month of an OCM team member's time could be saved, allowing them more time to spend on the more complex problems and not delay the simpler ones. Thus, the UFC Advisor could save considerable time just where the RAR is concerned.

In addition, because the M&I and RAR tests are so similar, the system is capable of supporting the M&I test. This is because the recommendations that the system makes are often concerned with the adjustments and replacements that could be made to bring testpoints into limits during an M&I. Since the M&I test is operator-intensive, any time savings would increase both test stand and operator availability considerably.

The design of the UFC Advisor centers around the linking of testpoint out-of-limits data with possible problems and then linking possible problems to possible solutions. These linkages are provided as static knowledge in the UFC Advisor. The dynamic knowledge in the UFC Advisor is then essentially a diagnostic inference engine, implemented in CLIPS, than can utilize the linkages to identify potential problems, prioritize the problems and solutions, and write a report containing recommendations on what to do next. This diagnostic inference engine is a very general tool that could be utilized in any knowledge-based system development effort that is to interpret testpoint data and provide recommendations. Only the static knowledge containing the information linking testpoints out-of-limits to problems and solutions would have to be changed to fit the new device being tested.

```
***************************************************************
*               UFC ADVISOR EXPERT SYSTEMS ANALYSIS           *
*                                                             *
*                             for                             *
*                                                             *
*                      FUEL CONTROL # 50340                   *
*                                                             *
***************************************************************

Summary of Test Points
(Points out of limits marked by '*')

Para  TP   Item                Min           Recorded      Max
----  ---  --------------   -------        --------     -------
66011 340  PLAP-DIFF           0.20           4.20*        0.80
66011 350  PLAP-DIFF           0.10           6.30*        3.00
66011 370  PLAP-DIFF           0.10           9.60*        3.00
12007 090  WF4              1245.00        1479.00*     1395.00
14005 010  WF4              1245.00        1454.00*     1395.00
15002 040  WF4              -250.00        -365.00*      150.00

Governor Problems...
  Troubleshoot the Governor Section and run GG Complete
  P 15002  tp 40  Item WF4  OOL low  RCRD -365
  P 14005  tp 10  Item WF4  OOL high  RCRD 1454
  P 12007  tp 90  Item WF4  OOL high  RCRD 1479


PROBLEM: Augmentor Computer

   EVIDENCE: P 66011 tp 340 Item PLAP-DIFF OOL high
               MIN 0.200    RCRD 4.200    MAX 0.800
             P 66011 tp 350 Item PLAP-DIFF OOL high
               MIN 0.100    RCRD 6.300    MAX 3.000
             P 66011 tp 370 Item PLAP-DIFF OOL high
               MIN 0.100    RCRD 9.600    MAX 3.000
   SOLUTION: Demate to augmentor computer and check for leaks or
             problems with the segment 5 solenoids.


PROBLEM: Idle Governor

   EVIDENCE: P 12007 tp 90 Item WF4 OOL high
               MIN 1245.000    RCRD 1479.000   MAX 1395.000
   SOLUTION: Recheck governor part power.  If on low side,
             adjust N2 cam follower.
   SOLUTION: Adjust PLA' trim cam follower and/or N2 request
             servo.
```

FIGURE 3. Example of a portion of the UFC Advisor's output