DART: An Expert System for Computer Fault Diagnosis

James S. Bennett
Heuristic Programming Project, Computer Science Department
Stanford University, Stanford, CA 94305

Clifford R. Hollander
IBM Scientific Center, 1530 Page Mill Road
Palo Alto. CA 94304

A. Introduction

We describe an application of artificial intelligence techniques to computer system fault diagnosis, in particular, we have implemented an automated consultant that advises IBM field service personnel on the diagnosis of faults occurring in computer installations. The consultant identifies specific system components (both hardware and software) likely to be responsible for an observed fault and offers a brief explanation of the major factors and evidence supporting these indictments. The consultant, called DART, was constructed using HMYCIN [1], and is part of a larger research effort investigating automated diagnosis of machine faults [2].*

B. Project Motivation and Scope of Effort

A typical, large-scale computer installation is composed of numerous subsystems including CPUs, primary and secondary storage, peripherals, and supervisory software. Each of these subsystems, in turn, consists of a richly connected set of both hardware and software components such as disk drives, controllers, CPUs, memory modules, and access methods. Generally, each individual component has an associated set of diagnostic aids designed to test its own specific integrity. However, very few maintenance tools and established diagnostic strategics arc aimed at identifying faults on the system or subsystem level. As a result, identification of single or multiple faults from systemic manifestations remains a difficult task. The non-specialist field service engineer is trained to use the existing component-specific tools and, as a result, is often unable to attack the the failure at the systemic level. Expert assistance is then required, increasing both the time and cost required to determine and repair the fault. The design of DART reflects the expert's ability to take a systemic viewpoint on problems and to use that viewpoint to indict a specific components, thus making more effective use of the existing maintenance capabilities.

For our initial design, we chose to concentrate on problems occurring within the teleprocessing (TP) subsystems for the IBM 370-class computers. This subsystem includes various network controllers, terminals, remote-job entry facilities, modems, and several software access methods. In addition to these well-defined components there are numerous available test points the program can use during diagnosis. We have focussed our effort on handling two of the most frequent TP problems, (1) when a user is unable to log on to the system from a remote terminal, and (2) when the system operator is unable to initialize the TP network itself. In a new system configuration, these two problems constitute a significant percentage of service calls received.

Interviews with field-service experts made it apparent that much of their problem-solving expertise is derived from their knowledge of several

well-defined communications *protocols*. Often composed of simple request-acknowledge sequences, these protocols represent the transactions between components that are required to perform various TP tasks. Although based on information found in reference manuals it is significant that these protocols are not explicitly detailed anywhere in the standard maintenance documentation. Knowledge of the basic contents of these protocols and their common sequence forms the basis of a diagnostic strategy: use the available tracing facilities to capture the actual communications occurring in the network, and analyze this data to determine which link in the protocol chain has broken. This procedure is sufficient to identify specific faulty components in the network.

C. The DART Consultation

During a DART consultation session, the field engineer focusses on a particular computer system that is experiencing a problem. Many installations arc composed of numerous CPUs that partially share peripherals, thus, the term "system" is taken to mean a single CPU-complex and its attached peripherals. Within each such system, the user describes one or more problems by indicating a failure symptom, currently using a list of keywords. Using this description, the consultant makes an initial guess about which of the major subsystems might be involved in the fault. The user is then given the opportunity to select which of these implicated subsystems are to be pursued and in which order.

Each subsystem serves as a focal point for tests and findings associated with that segment of the diagnostic activity. These subsystems currently correspond to various input/output facilities (e.g.. DISK. TAPE. TP) or the CPU-complex itself, lor each selected subsystem, the user is asked to identify one or more *logical pathways* which might be involved in the situation. Kach of these logical pathways correspond to a line of communication between a peripheral and an application program. On the basis of this information and details of the basic composition of the network, the appropriate communications protocol can be selected. The user is also asked to indicate which diagnostic tools (e.g., traces, dumps, logic probes) arc available for examining each logical pathway.

Once the logical pathway and protocol have been determined, descriptions arc gathered of the often multiple *physical pathways* that actually implement the logical pathway, it is on this level that diagnostic test results arc presented and actual component indictments occur. For DART to be useful at this level, the field engineer must be familiar with the diagnostic equipment and software testing and tracing facilities which can be requested, and, of course, must also have access to information about the specific system hardware and softwate configuration of the installation. Finally, at the end of the consultation session, DART summarizes its findings and recommends additional tests and procedures to follow. Figure 1 below depicts the major steps of the diagnostic process outlined above.

Report Findings and Recommendations
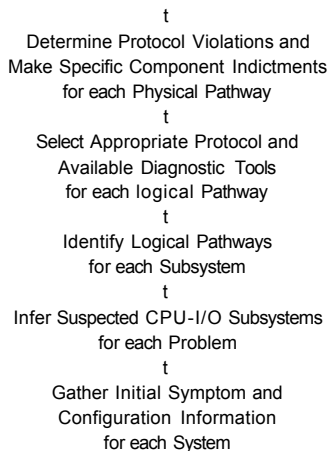for the Session

t

Determine Protocol Violations and
Make Specific Component Indictments
for each Physical Pathway

t

Select Appropriate Protocol and
Available Diagnostic Tools
for each logical Pathway

t

Identify Logical Pathways
for each Subsystem

t

Infer Suspected CPU-I/O Subsystems
for each Problem

t

Gather Initial Symptom and
Configuration Information
for each System

Figure 1. The DART Diagnostic Inference Process

The following transcript illustrates DARI's interactions with an engineer and DART's conclusions about possible faulty components. The engineers responses arc emboldened; annotations appear in *italics*.

. . . . .SESSION-201 . . . .
1) What Is your user-number?
•• 009776
2) Are you familiar with the use of this tool?
•• YES
3) Briefly describe the reason for this session:
•• HAVING DIFFICULTY ACTIVATING T*P* NETWORK

The first system for SESSION-201 will be called:
. . . . . .--SYSTEM-1---. . . . .
5) Is SYSTEM-1 a loosely coupled multi-processor?
•• NO
The first problem with SYSTEM 1 will be called:
. . . . . . .PROBLEM-1————
Please enter the following information.
CPU   SCP
machine   SCP   running
type   name   under VM?

SCP is the operating system type;
VM is an IBM-supplied virtual machine operating system

6) •• 3081   MVS/SP   NO
7) What symptom is exhibited by PROBLEM-1?
•• ??
Expected responses are: UNABLE.TO.LOGON, UNABLE.TO.ACTIVATE.TP.NEIWORK. MISSING.DATA, HANG. LOOP, etc.
•• UNABLE.TO.ACTIVATE.TP.NETWORK
The user employed KM YC/N's automatic completion feature

8) Is the NETWORK.ACTIVATION problem occurring in a SNA network? (YES NO)
•• YES   *SNA is a particular network discipline*
9) Is the terminal(s) involved in PROBLEM-1 locally (channel) attached, remotely attached or both? (LOCAL REMOTE BOTH)
•• REMOTE

At this point the TP subsystem becomes the focus of the diagnostic activity

The TP subsystem for PROBLEM-1 will be called:
. . . . . .--TP.SUBSYSTEM-1. . . . . . -

Complete one line for each remote terminal.
applic.   communic.
terminal   cluster   program   controller
LU-name   PU-name   LU-name   address
HI- logical unit, PU-physical unit

13) •• TERM1   CLUS01   IMS   069
14) •• DONE
The path to terminal TERM1 will be called:
————SNA.PATH-1————
15) What is the name of the LINE connecting CLUS01 to 059 (its TP controller)?
•• LINE01

18) Please enter the name of the access method used by IMS to communicate with TERM1. (ACF/VTAM VTAM ACF/TCAM TCAM)
•• ACF/VTAM
19) What type of TP controller is used for this terminal? (3704 3705)
•• 370G
20) What control program is running in this TP controller? (NCP PEP)
•• NCP

22) Please indicate ALL of the nodes which are able to achieve and maintain (throughout the activation process) an active status.
* • I MS *There is a console command a vailable which can be used to interrogate node status.*

To most effectively identify the cause of this problem will require a trace showing the activation of the LINE, the 3705 and NCP.
23) Which VTAM trace types arc available to aid in pursuing this problem?
•• VTAM.10
24) Does your VTAM 10 (RNIO) trace show that an ACTLINK (activate link) command was sent to line LINE01?
•• YES
25) What does your VTAM 10 (RNIO) trace indicate was the response to the ACTLINK? (POSITIVE NEGATIVE NONE)
•• NEGATIVE

The most likely source(s) of the failure discussed in PROBLEM-1 is:

LINE-LINE01   (6)
The VTAM l0 (RNIO) trace indicates that an ACTLINK was sent to NCP for LINE01 and that a negative response was received, The ACTLINK should have caused NCP to issue an ENABLE to the line and this, in turn, should have caused the modem to respond with DATA SET READY (DSR). You might want to check the modem interface, If the DSR is active then it is 1 ikely that the line is not the source of the failure. A LINE (or PT2) trace could be used to further investigate the problem.

3705-059   (.2) (See evidence for LINE01).

NCP   *(.2)* (See evidence for LINE-ITNFfm

844

After DART has indicated the components most likely to be at fault, the responsibility for performing a detailed determination and repair of the actual component failures (i.e.. microcode bugs, integrated circuit failure, etc.) would then shift to the appropriate maintenance groups for those components.

D. Concluding Remarks

The current DART knowledge base consists of 300 EMYCIN parameters and 190 production rules and was constructed over a period of 8 months. During this period 5 specialists were interviewed about different aspects of the diagnostic process and the knowledge base reflects their composite expertise.

As might be expected, much of the requested diagnostic data is already in a machine-readable form on the subject computer system. However, as the transcript shows, this information must currently be entered by the user. This interaction forms a substantial fraction of me users input. Indeed, wc estimate 30 to 60 percent of the current interaction between program and user will be eliminated when this on-line data is exploited.

It is clear that the communications protocols form the crux of the expertise, both for the human specialist and for DART. Although our experts were able to easily articulate these protocols, their translation into the production rule formalism was tedious. A protocol represents an expected sequence of transactions between components. However, in order to indentify specific faulty components, the production rules capture only the possible *deviaiions* from this expected sequence. Thus each protocol yields a substantial number of rules which only indirectly reflect the original sequence and tend to produce rather opaque explanations of the diagnostic reasoning. Furthermore, for any lengthy protocol, ensuring the *completeness* of the resulting ruleset becomes a significant problem. In a collateral effort, wc are investigating the use of explicit representations of these protocols with general diagnostic rules which will hypothesize deviations directly from the protocols.

Acknowledgements

References

1.   van Melle, W.. *A domain-independent system that aids in constructing knowledge-based consultation programs*. PhD dissertation, Computer Science Department, Stanford University, June 1980.

2.   Computer Science Department, *The Heuristic Programming Project" 1980*. Computer Science Department, Stanford I Iniversity, 1980.