EXPERT SYSTEMS TACTICS IN TESTING FASTBUS SEGMENT INTERCONNECT MODULES (LESSONS SO FAR IN THE SIDES PROJECT)

R. N. Knowles, R. Forster, N. Ho, and J. Kroma
*Fermilab National Accelerator Laboratory
P.O. Box 500
Batavia, Illinois 60510

The Fastbus Segment Interconnect (SI) module is among the most complex modules to diagnose and repair. The SI, being an intermediary device, provides a temporary connection between two otherwise independent Fastbus segments. Internal SI problems typically manifest themselves elsewhere within the segments served. Off-line testing of this sophisticated device is equally challenging, and it's into this environment that Fermilab has launched an exploratory project called the SEGMENT INTERCONNECT DIAGNOSTIC EXPERT SYSTEM, (SIDES), to evaluate the technology as well as to create a more effective and streamlined testing schema. Presented here are some of the issues which have arisen during the initial phases of this project.

1.0 Sides Background

Fermilab has participated in all phases of Fastbus's development and implementation (IEEE 960-1986). It was with some eagerness that it received the first Segment Interconnect cards. These initial units were checked out by a highly experienced engineer, and once working, were immediately placed into service. As there were only a few, and as they were not yet accepted as being adequately debugged, there was no great wave of demand at first. Slowly however, the demand increased, and it was apparent that the testing of these devices would need to be transferred from the engineering setting into the realm of the lab's Instrument Repair Group in order to cope.

The testing employed the use of a large complement of computer equipment, consisting of several Fastbus Segments (Backplane and Cable), two Digital Equipment Corporation PDP11 type computers, and the associated DRV11J/IORF1 host interfacing. These systems need to be configured and reconfigured during testing into the various hook-ups needed to test certain facets of the SI.

Previously, a large library of diagnostic test programs were developed by the Colliding Detector Facility in support of its own Segment Interconnect support needs. These programs were implemented in FDL (Fastbus Diagnostic Language, a specialized, Pascal-like diagnostic language focussing on FASTBUS, first introduced by Dave Lesny of the University of Illinois and modified by Fermilab and several other institutions.) These tests, although comprehensive, execute slowly, and of course, as with any complex software package, demand much from the technician in the way of understanding and interpreting their error returns.

Early conceptions of SIDES envisioned an expert system linked intimately with these same diagnostics, thereby capitalizing on the existing expertise encoded within them.

*Operated by Universities Research Association under Contract with the U.S. Department of Energy.

2.0 Sides Implementation Issues

2.1 Is an Expert System Even Called for?

Before discussing what has been learned and accomplished so far in pursuit of SIDES, it is appropriate to lightly explore the more general issue of expert system applicability to a given problem. [1] There are 3 issues to be considered:

- o Do experts on the target subject exist?
- o Can they solve these problems?
- o Can they solve these problems remotely? (The Telephone Test)

If experts do not exist, then there is no source for the expert heuristics, and therefore, no point in attempting an expert system.

In the case of SIDES, all three tests were applied with positive results, experts did exist, they could solve such problems and such expertise could easily be delivered remotely.

2.2 Understanding the Problem Domain

Having determined that an expert system could be built, the next step was to survey the problem domain to attempt to understand what sort of time and resources would be required in such an undertaking. A number of different approaches were tested before finally arriving at a reasonable model. Tests on expert systems characterize these two basic approaches as the Entity-Relationship model and the Hierarchical-Reductionist model.

With the Entity-Relationship model, one attempts to define the problem's elements, their interactions and the extent of such interactions. Primarily employed in the construction of relational databases, it was discarded as being too cumbersome an approach.

SIDES H-R Diagram

Figure 1.

More satisfactory results were realized when the Hierarchical-Reductionist model was employed. In this approach, the problem domain is broken into successively smaller elements, until one arrives at a level which either is indivisible or at which it loses its relevance if divided further (i.e. dividing an automobile into iron and chromium atoms is hardly relevant to diagnosing ignition failures). Figure 1 demonstrates a small portion of the Segment Interconnect which as been rendered into this form.

Sadly, as work has proceeded with the design, this "neat and clean" form was discovered to be unrealistic, since there are some circumstances where the devices sub-elements interact, somewhat complicating the model. For the most part however, such a breakdown seems to suffice for purposes of generally understanding the domain.

2.3 Shell Construction

Remaining consistent with imposed project limitations, the early SIDES designers chose to employ the OPS5 language as the basis of the shell. OPS5, first envisioned by Charles Forgy in the 1970's is a Rule-Based production system, lending itself well to the IF..THEN..structure common to expert system rules. Further enhancing its attractiveness to SIDES is the fact that the language is available and supported by DEC on their VAX family. Incidentally, is also used extensively by DEC on their XCON system, that company's own internal VAX hardware configuration system.

Contemporary expert systems typically consist of a vendor-supplied shell program, into which the knowledge engineer encodes his/her rules. As the DEC OPS5 product has no such support, it has been necessary to create such a shell to suit our purposes. This is just as well as no commercial shell could be easily merged with the intended test-program library mentioned above.

Having reduced the problem domain into bite-sized chunks, it was felt that the shell structure would have far-reaching implications as to the way the expert-system rules would be structured. As such, pains have been taken to organize it in a complementary manner.

SIDES Expert System Shell

Figure 2.

Figure 2 illustrates the main control flow of the shell. Once initialized, the expert system begins execution at the "Find out about" (FOA) which calls the appropriate fortran test/subtest. The requested test, having completed, builds a parameter passing construct called a Result Element, which it then passes back to the OPS5 shell, implicitly reactivating it.

The SCOPE block acts to recognize such returns, passing control to the SCHEDULE block should no error be in evidence, or activating the EXPLAIN block should an error condition exist.

EXPLAIN, containing the bulk of the expert system's heuristics, attempts to pattern-match the error condition. It then creates internal control parameters which are utilized by SCHEDULE to provide a next-test schema which is again employed by FIND_OUT_ABOUT.

The above loop is driven by what OPS5 terms a Vector Driven Agenda, finding its chief advantage in being able to dynamically manage its execution, making changes when necessary, but following a pre-defined goal-driven flow.

A further reason for attempting to control the execution of the shell is to allow for what is referred to by DEC as "Context Limiting", reducing the number of rules enabled at a given time, thereby speeding execution and simplifying debugging. [11]

2.4 Acquiring and Representing Expert Knowledge

Many issues in this respect are anecdotal. Volumes have been written exploring this subject in great depth. [1,2,3,12,13] From our own experience we have discovered a few counter-intuitive aspects which are worth mentioning.

Studying the transcripts of our first interview efforts leads us to the conclusion that they were feeble indeed! It has become painfully clear that without a directed agenda, most expert interviews deteriorate into pleasant chats, providing few usable results.

Having a hard agenda implies that the knowledge engineer should have a fairly clear idea of what areas are to be focussed upon, hence the need for a good 'road-map' such as the H-R chart of Figure 1 to work from.

Taping and transcribing the interviews provides us with a hard-copy document which could be examined at length by the project's participants. This transcription is then incorporated into the project's archives for future reference.

Another source of expert information for the project has been the FDL diagnostic test programs mentioned previously. Again, our principle problem in coping with this expert data is finding the appropriate representation, and efforts continue to explore the most effective format.

Finally, as work proceeded it became apparent that there were some testing scenario's where an expert system shell was simply too cumbersome. Recognizing this, we have crafted the overall system to allow for both "non-expert", direct driven testing as well as the expert system. These are linked by a

menu choice, allowing the technician the freedom of doing testing such as scope loops. Figure 3 depicts this "dual channel" approach.

SIDES Overall Structure

Figure 3.

3.0 Sides Goals/Futures

Most of the future goals outlined in [8] still stand but I reiterate them here for reference.

"MORON" Level Diagnostic Driver/Expert Pattern Matcher:

Our fledgling system currently is able to call the external test programs, recognize error returns, and respond (in a rudimentary fashion) to them. This interface is being further expanded to work efficiently with our dual-channel scheme, and we expect by summer 1989 to have all tests within reach of the expert shell.

EXPERT Explanations:

Although some very crude explanations are now being done, we hope to incorporate a much more robust capability in subsequent versions of SIDES. Many of the mechanical portions will be relegated to external routines. The expert system will then pass selected bits of information which will then be incorporated into the full explanation. Currently we are ignoring this dimension, concentrating our efforts on making SIDES's primary functions work.

Expert Test Planner/Diagnoser:

As an expert diagnostic system, SIDES will have to be able to manage the full suite of diagnostic tools available to it. This then encompasses the truly "expert" aspect of the system. As such, much of the interviewing accomplished and planned is directed towards capturing those Heuristics. This will no doubt continue through the project, and besides being a current goal, also constitutes the bulk of design work yet to be done.

One of the strengths of OPS5 is its lack of context sensitivity, and consequently, we will be able to expand the knowledge base easily, writing new rules well into the maturity of the project.

Expert Testing Associate:

The most aggressive goal of SIDES is to extend its interface to actually interacting with the technician, aiding him with testing and troubleshooting, suggesting strategies, and providing him with an "expert's insight". We are far from this goal at present.

Conclusion

Besides providing an interesting challenges, the SIDES will yield many answers pertaining to the efficacy of expert systems in the FERMILAB environment. Many possibilities exist for the application of this technology in such areas as more advanced Expert Control and Diagnostic Systems such as the FLARES system [5], Expert Task Advisors [10], and Expert Computer Tuners [9].

References

- [1] Hayes-Roth, Fredrick et.al., "Building Expert Systems" Addison-Wesley Publishing Co. ISBN 0-201-10686-8.
- [2] Winston, Patrick Henry, "Artificial Intelligence" Addison-Wesley Publishing Co. ISBN 0-2-1-08259-4 (Chapter 6 especially).
- [3] Nilsson, Nils J., "Principles of Artificial Intelligence" Morgan Kaufmann Publishers Inc. ISBN 0-934613-10-9.
- [4] Aiello, Luigia, et.al., "Representation and Use of Metaknowledge" Proceedings of the IEEE, Vol. 74 No. 10, October 1986.
- [5] Bill-Wray, Susan E. et.al., "The Implementation of the Fault Localization, Assessment and Repair Expert System (FLARES)--Tools, Languages and Issues", Proceedings of the Digital Equipment Computers Users Society, December 1985.
- [6] Wise, Rebecca, "Knowledge Representation Issues", Proceedings of the Digital Equipment Computer Users Society, December 1986.
- [7] Georgeff, Michael P. and Amy L. Lansky, "Procedural Knowledge" Proceedings of the IEEE, Vol. 74 No. 10, October 1986.
- [8] Booth, W. Alexander, et.al., "SIDES-Segment Interconnect Diagnostic Expert System", IEEE Transactions on Nuclear Science, February 1989.
- [9] Small, Robert A., "Development of a VAX Tuner Using OPS5", Proceedings of the Digital Equipment Computer Users Society, Spring 1986.
- [10] Dhar, Vasant. Pople, Harry E., "Rule-Based Versus Structure-Based Model for Explaining and Generating Expert Behavior", Communications of the ACM, June 1987, Vol. 30, No. 6, PP542.
- [11] Brownston, Farrel, Kant and Martin, "Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming", Addison-Wesley Publishing Company, Inc., 1985.
- [12] Tanimoto, Steven L., "The Elements of Artificial Intelligence", The Computer Science Press.
- [13] Cooper, Thomas A, and Wogrin, Nancy, "Rule Based Programming with OPS5", Morgan Kaufmann Publishing.