SSCFI: Autonomous Fault Isolation in Communications Circuits

Ralph Worrest, Roland Zito-Wolf[1], Hongbin Wang, Shri Goyal
Integrated Intelligent Systems Department
GTE Laboratories, Incorporated
40 Sylvan Road
Waltham, MA 02254 USA
rworrest,rzito-wolf,hwang,sgoyal@gte.com

Abstract

SSCFI[2] is a rule-based expert system that diagnoses problems in a wide variety of "special circuits," that is, telephone circuits other than regular switched business and residential lines. Special circuits are significantly more complex than regular circuits, and hence more difficult to diagnose. SSCFI diagnoses problems by recursively partitioning the circuit until the responsible fault is isolated. SSCFI selects which circuit to work on, reads its design, selects and initiates analog and digital tests via remotely-activated test equipment, interprets the results of each test in the context of the circuit design, and when done, writes out a detailed description of the problem found and routes it to the party responsible for its repair. SSCFI is entirely autonomous in operation.

SSCFI has become an essential element of GTE's special circuit maintenance operations. It has been in service since 1991 and has been in operation at all GTE's US sites since 1994. SSCFI testing saves millions of dollars annually and significantly improves the uniformity of testing and quality of the resulting diagnoses.

This paper discusses the domain, architecture, and development of the SSCFI system, and the key factors and techniques that made it successful. Lastly, two current projects building on SSCFI's expertise are discussed -- interactive test assistance and automatic design database cleanup.[12]

Problem Description

GTE is a major provider of telecommunications services, with over 20 million customer circuits. Of these, roughly 1 million are "special circuits," which includes any telephone circuit except a normal residential or business connection -- bank ATM, foreign-exchange, off-premise extension, high-capacity, hard-wired, or otherwise customized circuits (GTE, 1990). Special circuits are significantly more complex than regular circuits and typically span multiple central offices (COs). Maintenance of these circuits is a significant problem because locating faults in geographically extensive circuits is labor-intensive and slow -- possibly requiring the cooperation of technicians at multiple central offices as well as repair personnel in the field -- and because repair time requirements for special circuits are significantly more stringent than for regular circuits. Consequently, GTE has equipped many special circuits with remote testing capability, allowing circuits to be diagnosed from centralized testing centers.

When a customer reports trouble on a special service circuit, the customer service representative enters the raw problem data into a workflow system, the Trouble Administration System (TAS). A *trouble ticket* is created which represents that problem. The ticket is then routed to a Special Service Control Center (SSCC). SSCC personnel test circuits using several remote test systems, including SAS™, SARTS™, AUTOTEST 2™, and REACT 2000™. The tester first verifies the reported problem and then isolates the fault as much as possible, ideally localizing the fault to a specific location to which a technician can be dispatched. He then writes his observations and conclusions onto the TAS trouble ticket and instructs TAS to dispatch it to (that is, to place it in the work queue of) the party responsible for repair or further diagnosis. Possible dispatch locations for faults include: a central office, the outside plant associated with a specific CO, and the customer premises.

Fault isolation is a kind of diagnosis, differing in that the primary goal is to isolate the fault to within a particular organization's area of responsibility (e.g., a particular CO), rather than to a particular faulty component. This reflects the practical tradeoff that while it is important that the tester make as specific a diagnosis as possible, it is even more important that the ticket be routed as quickly as possible, which implies that the number of tests performed be minimized.

---

[1] Address correspondence to: rzito-wolf@gte.com, (617)466-2470, FAX (617)466-2960

[2] Pronounced "scuffy"

Automated Fault Isolation

While remote testing significantly improved the repair process, the large volume of work at the SSCCs required several hundred test personnel and was a major expense. There were also problems with testing quality, due to the wide variation in tester expertise and the large number of special circuit configurations, and with testing time, due to the need to access multiple systems per diagnosis. Reducing testing time is especially important because over the past few years the allowable time-to-repair has been reduced from 4 hours to 1-3 hours, depending on the circuit type.

GTE's answer to this problem was to automate the fault isolation process. SSCFI (Special Service Circuit Fault Isolation) diagnoses troubles reported in a wide variety of GTE special service circuits. SSCFI operates similarly to a human tester in the SSCC environment (see Figure 1). SSCFI polls the TAS work queues for trouble tickets, selecting the highest-priority trouble among the unassigned work. If additional design information is required, SSCFI accesses the CNAS II design database to obtain it. SSCFI then invokes one or more remote test system(s) to verify and isolate the fault. Lastly, SSCFI writes a summary of its conclusions (its *remarks*) onto the TAS ticket and dispatches it.

SSCFI is a model-based expert system (Davis & Hamscher, 1992). It reads the target circuit's design to generate an internal circuit model; it then selects tests with the goal of maximizing diagnosis quality and minimizing test time. SSCFI has specialized knowledge about circuit types, testing, and diagnosis, and can currently test most types of special circuits, both analog and digital. Unlike many expert systems, SSCFI operates *on-line* and *autonomously*. SSCFI is responsible for determining when it cannot successfully test a circuit, perhaps due to an unsupported configuration or lack of test access, and referring it to a human tester. SSCFI is able to recognize test-equipment and other system-level problems and page an appropriate human to get them resolved.

One current limitation is that SSCFI must rely on the automated systems for all its input, whereas human testers often work cooperatively with field personnel to resolve difficult troubles. When SSCFI cannot access a circuit, requires a test assist, or cannot satisfactorily isolate a fault, it dispatches the trouble to SSCC human personnel. In such cases, SSCFI attempts to summarize in its remarks whatever results it has been able to obtain, to help whomever next works on the ticket.

SSCFI has become an essential component of GTE's special circuit maintenance operations. It has been in operation at all GTE's special-service testing centers since 1994. SSCFI currently performs more than 40,000 circuit diagnoses per year; this number will continue to rise as SSCFI's knowledge of circuit types, area-specific practices, and testing methods is expanded.

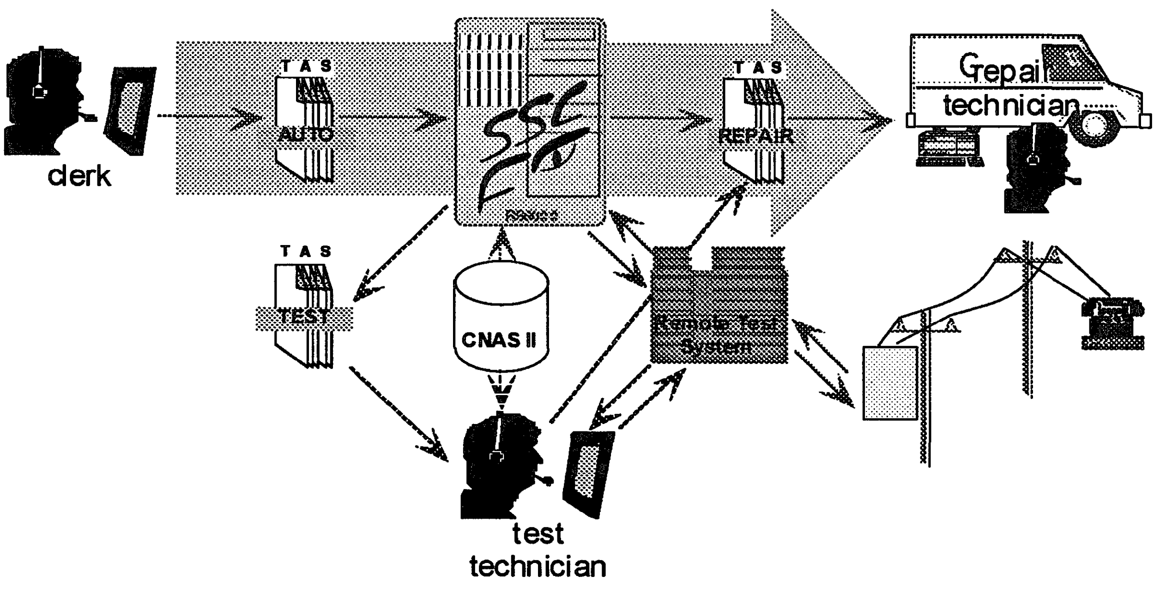

Figure 1: SSCFI's Work environment

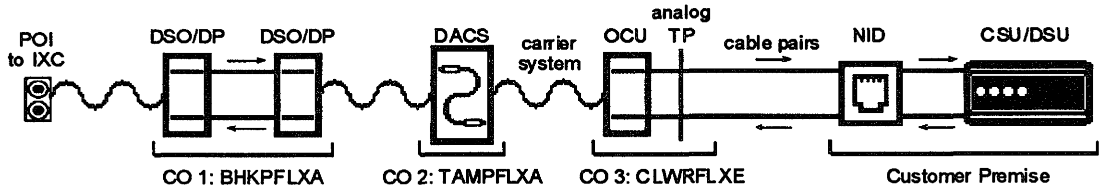

Figure 2: A Typical Point-to-POI Digital (DDS) Circuit

Problem Example

Figure 2 illustrates a typical low-speed digital (DDS service) circuit. The left side is the *A-end* or *top*; the right side the *Z-end* or *bottom*. At the A-end is a POI, a *point of interconnect*, indicating that beyond this point the circuit is provided by another telephone company. At the Z-end is a CSU/DSU (a customer's modem), making this a *point-to-POI* circuit. (When both ends of a circuit terminate at customer equipment, it is called *point-to-point*.) From the POI to the Office Channel Unit (OCU), the circuit is multiplexed onto high-bandwidth digital carrier systems. A signal can pass through multiple offices on digital carrier systems. Carriers can be connected at intermediate offices through a digital cross-connect system (a DCS) as in TAMPFLXA, or via back-to-back channel units (DSO/DPs) as in BHKPFLXA. DCS interconnects provide digital test access to the circuit. From the OCU in CLWRFLXE to the CSU/DSU at the customer's site, the circuit is a dedicated 4-wire circuit (transmit and receive *pairs*). An analog test point is generally included in this section to allow analog tests of each pair.

In digital circuits each piece of equipment has an addressable identity and can be individually looped back, that is, put into an echoing mode to verify signal transmission to the device and back. In this circuit, it is possible to separately loop the OCU, the NID, the CSU, the DSU, and each DSO/DP, all from the DCS test point. Faults are isolated primarily through differential loopback tests, plus analog measurements on the local loop. In contrast, analog-circuit fault isolation relies about equally on loopbacks and continuity tests.

To make it easier to distinguish between failures in GTE's and the customer's equipment, GTE generally provides a loopable *network interface device* (NID) -- at the customer site. If a tester can loop the NID but not the CSU/DSU beyond it, then the problem is most likely the customer equipment; if he cannot loop the NID, its a GTE fault. Without a NID, it is difficult to remotely distinguish between GTE and customer problems.

This domain has several challenging features:

- There is great variety in equipment behaviors and circuit configurations.

- The circuit design is generally not fully known, as the design records are neither complete nor fully reliable. For example, it happens that DDS circuit designs are unreliable as to whether a NID is present. This information is important when the tester is unable to loop any equipment at the customer's site. If the tester can't be sure a NID is present, then field personnel must be dispatched to the customer in many cases that are actually the customer's problem. The human testers determine if a NID is present (when it fails to loop) by the impedance signature observed from *analog* testing.

- The test equipment can be misconfigured or unreliable. Analog test points are often wired with the pairs swapped or the ends of the circuit reversed. Testers must recognize such conditions and compensate.

Why a Rule-Based Approach?

This domain has several features that suggest a rule-based approach.

- *Much of the experts' knowledge is procedural* -- situation-specific rules such "when you see an X fault in local circuit configuration Y, do test Z" -- which are highly amenable to expression in rules. The experts' primary diagnostic method is successive division of the circuit based on simple causal knowledge. They also use significant amounts of heuristic knowledge about the properties of specific circuit types and components.

Rule-base programming allowed us to express the experts' diagnostic procedures directly. It facilitated the construction of an initial system and incremental expansion of its competence. It also turned out to be a major factor in making the system's operation and results *understandable* – to both the technicians who implement SSCFI's repair recommendations and the testing experts who evaluate them.

• *Detailed circuit modeling is not required.* The testing experts generally do not know all the details of the underlying technology, but instead make do with a general models of each class of component. The component modules are purchased from third parties, meaning that their internal operation is generally proprietary, poorly-documented, and subject to change. In addition, the circuit model isn't fully known beforehand. The tester's model of the circuit can change as testing proceeds and additional evidence is accumulated.

• *The primary goal of testing is to get the circuit back into operation.* It is desirable but not necessary that testing be optimal or that the diagnosis be exact. For example, it is faster to replace suspect components than to try to identify a precise fault etiology.

SSCFI Architecture and Operation

SSCFI runs on RS6000 workstations under the AIX™ operating system. Each RS6000/580 supports 5-6 testers running simultaneously, plus various daemon processes handling administrative monitoring and control for the testers on that machine. Each SSCFI tester is capable of handling about 10-12 tickets per day.

Each tester is composed of two processes, the diagnostic process (the Knowledge Base, or KB, process) and a communications control process (the COMM process), which communicate via shared files and semaphores. The KB process controls the diagnostic session. It contains the knowledge about interpreting circuit designs, running tests, isolating faults, and describing results. The KB process is written in Brightware Corp.'s ART-IM™ (Brightware, 1988), a rule-based language. It currently includes about 1200 rules, 600 facts, 900 initial data structures, and 1600+ functions.

The COMM process is comprised of ExpecTerm scripts for requesting test operations and gathering data from each of the systems that SSCFI interfaces with. Expect (Libes, 1991) is an extension of the Tcl scripting language (Osterhout, 1994) for communicating with interactive processes; ExpecTerm is a further extension for interfacing with screen-oriented protocols. The COMM process's task is to manage the details of interaction with systems having terminal-oriented interfaces; it incorporates a minimum of testing and testing systems knowledge and so will not be discussed further here.

Diagnostic Algorithm

SSCFI's diagnostic procedure is outlined in Figure 3. The basic data structure used to control diagnosis is the fault-containing section (FCS). Each FCS specifies a fault observation and the section of the circuit within which the fault occurred. FCSs are used to reason explicitly about the problem-solving state (cf. NEOMYCIN: Clancey, 1988).

1. Select a ticket to work on. SSCFI prioritizes the pending tickets based on class of service and commitment time. An initial FCS covering the entire circuit is created for the reported fault.

2. Fetch and parse circuit design. SSCFI uses its design knowledge to fill in missing information and to check the design for consistency.

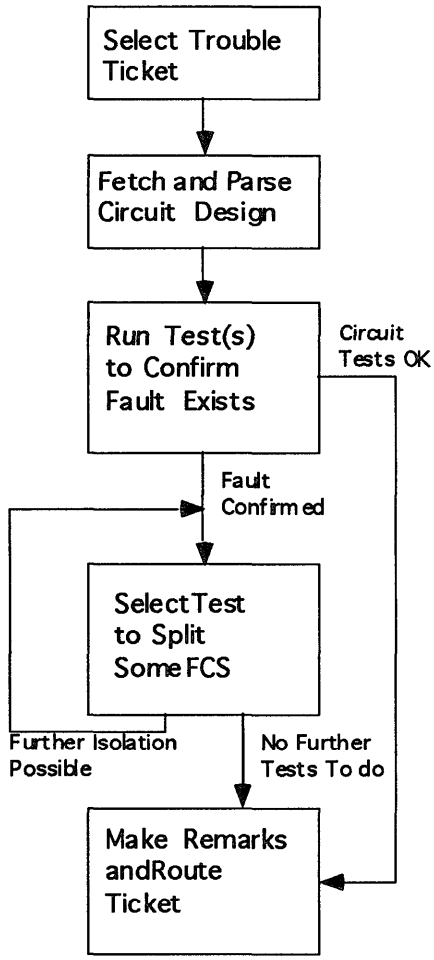

Figure 3: SSCFI's Diagnostic Algorithm

```
Auto results: LOOPBACK-FAILURE to A-END DSO/DP 2.

The reported trouble (DATA FAILURE) could be explained by:
  {1} LOOPBACK-FAILURE to A-END DSO/DP 2;
  {2} LOOPBACK-FAILURE to Z-END NID.

Discharge Summary:
  From SSCFI to BHKPFLXA: Please check {1},
    then route to CLWRFLXE (work loc: ACSC, dac: 8004).
  From SSCFI to CLWRFLXE: Please check {2}.

The above remarks are based on testing from the A-end DSO/DP 2 to the Z-end NID.

Summary of known good sections of the circuit:
  {4} LOOPBACK-OK to A-END DSO/DP 1 from DACS 1264-24 in TAMPFLXA
  {5} LOOPBACK-OK to Z-END OCU from DACS 1264-24.
```

Figure 4: SSCFI's Diagnostic Remarks for the Example Problem

3. Confirm the fault. SSCFI selects an initial test or set of tests that exercises as much of the circuit as possible, based on the type of circuit, availability of test access, and the type of trouble reported. If the circuit tests OK, the ticket is routed to a customer representative or a test technician; else SSCFI tries to isolate the fault(s).

4. Isolate the fault. While some unexplored FCS F exists, do:

- Select and run a set of tests that can potentially split F into sub-regions;
- Interpret test results and create a new FCS for each fault observed;
- Determine if any new FCS explains F;
- If F is now explained or no further subtests exist, mark F explored;

SSCFI performs tests to split each FCS until no further split is possible and worthwhile. SSCFI has knowledge suggesting appropriate tests to run to isolate each fault type, given the local circuit context and conditions; how to interpret the readings generated by the test and identify new faults; how to recover from bad or inconsistent test results, and so forth.

5. Generate remarks. The last step is to write out a description of the faults found and any other related information, and to route the ticket. Related information includes any additional information that might help localize the fault further, a statement of what part of the circuit was covered by the tests performed, and any miscellaneous observations or problems encountered during testing, such as test system problems, and non-explaining or minor faults.

Diagnosis Example

We now illustrate the diagnostic procedure in the circuit of Figure 1. In this example, both the farthest DSO/DP and the NID at the customer interface are faulty. (For brevity, details of fault selection and design interpretation are omitted.)

Confirming the fault. SSCFI first selects a test access point. Since this is a point-to-POI circuit, the DCS closest to the POI is selected; if it were point-to-point then any DCS point would be acceptable. SSCFI verifies the configuration of the DCS and takes a data sample; certain data codes are diagnostic of test equipment failure. If the DCS were misconfigured or faulty, SSCFI would notify the system administrator and look for an alternate DCS point.

SSCFI then confirms the fault by looping the end equipment in each direction. This is done with the "DDS macro" provided by the REACT test system, which performs loopbacks of all the customer-end devices -- the OCU, NID, CSU, and DSU -- in one operation. In this circuit the NID is faulty, so the DDS macro test in the customer (Z) direction returns "good" to the OCU and "failure" to all farther devices. In the A direction, the DDS macro fails to loop anything.

Fault Isolation. A latching loopback in the A direction to the furthest DSO/DP is attempted and fails. SSCFI then tries to loop back the nearer DSO/DP, which succeeds, meaning that the furthest DSO/DP has failed. Toward the Z-end, the differential between the loopbacks to the OCU and NID indicates that the fault must be contained in the section of the circuit from OCU to NID. There is an analog test point in this FCS, from which SSCFI performs analog measurements (voltage, resistance, and capacitance) in both directions. The resistance across each pair is in the 1K range, and the capacitances are normal. This indicates that equipment is connected and that the pairs are good to the NID, reducing the FCS to the NID itself. At this point, no further tests suggest themselves, so fault isolation is complete.

Remarks Generation. SSCFI's findings for this circuit are shown in Figure 4. The first line states the primary fault to be repaired. The second paragraph lists all faults found that explain the reported trouble and were not a consequence of some more specific fault. (If incidental faults were found, that is, faults that did not account for the reported trouble, they would be listed separately.) The third section indicates in detail how SSCFI suggests the ticket be routed, here, to the DSO/DP failure first and then to the NID failure in the field. Expressions in braces are references to previously mentioned test results.

SSCFI then explicitly states what portion of the circuit it believes it has tested, and lastly lists results documenting the part of the circuit that tested OK. Information beyond the primary diagnosis is provided for several reasons: it increases user confidence in the reliability of SSCFI's diagnoses, it makes the results more useful in complex cases such as multiple faults, and it facilitates retesting after repair.

Development History and Status

Initial knowledge acquisition for SSCFI began in September, 1989, focusing on analog testing in the South (Florida) region. It involved several weeks of expert interviews, from which a design and initial implementation were generated. The knowledge was then intensively refined for several months through expert review of SSCFI's performance on real cases. Only in this way could we elicit the tacit knowledge used in performing the task, knowledge that experts typically do not think to mention unless asked. Knowledge tuning continued for another 6-12 months of prototype operation until the knowledge update rate leveled off.

The first operational system was completed in May 1991, for selected voice circuits; full testing of analog voice and data circuits was achieved that October, followed by a significant period of tuning and further knowledge acquisition. SSCFI was extended to operate in the West (California) region in 1993, and countrywide by mid-1994. This effort involved a significant restructuring of the test-request mechanisms to accommodate an additional test system (SAS). Digital testing knowledge acquisition began in mid-1994, involving 3 weeks of interviews, 6 months of intensive refinement, and 6 months of tuning in the field. Testing of DDS-class circuits was operational by mid-1995.

Over the last three years significant improvements in the knowledge, efficiency, and success rate of the system have been achieved. SSCFI's success rate is now about 90%, up from 65% in 1992, and continues to improve; average test time has been reduced from 35-40 minutes to around 25 minutes under full system load. We expect to reach 20-minute testing in 1996 though hardware upgrades and replacement of the current terminal-emulation-based interfaces to external systems with program-to-program interfaces (APIs).

Maintenance

SSCFI's knowledge base is constantly evolving. Expert users perform regular reviews of cases with incorrect diagnoses, failed test requests, or excessive test time. This generates a steady stream of "bug" reports (about 1 per day). Of these, about 25% are due to new situations of various kinds -- design syntax variations, new or changed test system error messages, or unusual circuit configurations. Another 25% are minor enhancements worth doing as time permits. In addition, detailed reviews are conducted yearly. New releases are generated every 4-6 months.

Release testing is a major issue. Our release process includes a regression suite of over 200 cases, with more being added all the time. Regression testing has been very effective in exposing bugs and errors in the code.

There are several reasons for the continued knowledge base changes. One is that there is a lot of minor variation among cases and new variations are always turning up. This tends to level off over time. A second reason is that the users keep coming up with ideas for improvements, which we try to incorporate as much as possible. A third reason is that the domain is in constant flux. We are regularly confronted with changes in circuit equipment behavior, test system interface operation, and operational requirements such as workflow policies. This is the most serious maintenance problem.

Evaluation

SSCFI is regarded by the SSCC as a major success. SSCFI currently handles about 90% of the auto-testable circuits reporting troubles each day -- over 40,000 trouble reports per year. (Auto-testable circuits are those with remote test access, about half of GTE's special circuits.) When SSCFI is unavailable in a testing region, the SSCC testing staff are hard pressed to cope with the workload. Development required about 25 man-years; over the next 5 years, SSCFI is expected to return at least 7 times its development cost.

SSCFI is also recognized for significantly improving the overall quality of specials testing. SSCFI receives high marks from experienced testers for its thoroughness of testing and clarity of explanation. With the speed enhancements mentioned above, SSCFI will in most cases be faster than human testers as well, an important factor in the increasingly competitive telecommunications marketplace. The success of the system has prompted other organizations to develop automated specials diagnosticians, but to our knowledge no comparable system is currently available, commercially or otherwise.

Lessons Learned

Why is it Successful? There are several reasons for SSCFI's success.

SSCFI's domain and task environment are well-suited to a heuristic approach. The domain is one where the experts rely more on heuristic procedures derived from experience and simple causal models than on detailed knowledge of component behavior, which is varied and constantly changing. Computationally, the task is not too complex – a divide-and-conquer approach works in the majority of cases.

There is a wide distribution of skill levels among the testing staff. The term "expert system" is something of a misnomer: SSCFI has been successful in its target task because it brings a uniform and reliable level of competence to the testing task, rather than because it has achieved strictly "expert" performance.

Errors are not fatal in this domain. SSCFI makes accurate diagnoses most of the time, and can recognize and route most circuits it cannot handle, but it is only a program and errors still occur. The impact of diagnostic errors is limited because the SSCC's repair workflow separates diagnosis from repair. Every tester's diagnosis and routing is subject to revision by the technician in the field. This is a major reason why SSCFI can run autonomously. Running in an "advisor" or "assistant" mode would require that humans remain in the loop and would substantially reduce the cost benefits of the system.

No sophisticated user interface was required. SSCFI runs autonomously, so it was possible to focus on the core competence without being sidetracked by the substantial issues of providing user-friendly interfaces, of interactive explanation, and the like. Now that the system's value is established, we intend to explore the added value of interactive testing. The key point here is that the current interest in interactive operation is motivated by the tangible value of providing greater access to SSCFI's proven expertise, rather than the speculative one of providing a clever "assistant."

Significant effort went into maintaining sponsor and user interest. Developers maintained constant contact with both groups. The importance of this point should not be underestimated. The domain is one of continual small changes. The developers need to be aware of changes in technology, operational practice, management, and policy, preferably in advance. There need to be regular reviews of the system with the operational people involved to be sure that their needs and expectations are met. Several other development projects of a similar sophistication in the authors' experience failed to be established as an integral part of operations because they were unable to adjust their goals and schedules in response to such changes in operational requirements.

Managing Large Rule-Bases. The distributed structure of rule-bases facilitates incremental development but confounds modular design. Extensive use was made of state variables to partition and sequence rule subsets along functional boundaries; without this modularization the system would be unmaintainable. State sequencing also provides a simple way to implement closed-world assumptions ("if no rule has yet concluded X, then conclude Y") and replaced many uses of the computationally expensive "logical" construct (assertions which maintain their dependency information).

A second critical issue is optimization, which was essential to the viability of SSCFI. The literature on rule-base optimizations tends to focus on the join section of the RETE network (e.g., Giarratano & Riley, 1993; Brightware, 1988) where extremely costly errors are possible. Our experience is that, except for the occasional gross blunder, the greatest gains came from optimizing pattern (alpha) nodes (cf. the "average growth effect," Acharya, 1994) and reducing the number of individual RETE update calls (e.g., by batching updates).

What is the Task Expertise? As has been the case for many other "expert" systems, a significant portion of

SSCFI's knowledge relates to other than its nominal area of expertise, that is, diagnosis. The largest such body of knowledge regards the parsing and structure of designs. SSCFI routinely needs to adjust for erroneous or missing data and to infer missing components, such as the NID problem discussed above. Other problems include missing test points, incorrect test benchmark data, and undocumented equipment substitutions.

A second area of knowledge regards the testing equipment, which is not always reliable or properly configured. SSCFI had to be taught to distinguish failures of the testing process itself from those of the circuit under test, as the human testers do. This yielded a significant incidental benefit -- by routing explicit notifications of these problems directly to system administrators via their pagers, SSCFI has improved the testing environment for both human and automated testers.

Future Directions

SSCFI development is a ongoing process. A maintenance organization has been engaged to take over the ongoing tasks of user support, bug fixes, policy changes, release testing and management, and minor enhancements. This will enable us to focus on several significant enhancements in the coming year: extension to higher-speed digital services (T1/T3), continued knowledge tuning and enhancement, and direct (API) interfaces to the workflow and testing systems.

Perhaps the most significant change will be the implementation of an interactive interface to SSCFI which will allow it to be used on-line as either a diagnostic expert or an intelligent repair assistant, in addition to its current autonomous mode. Interactive operation will extend the system's usefulness in several ways: it will give SSCFI access to additional observations about circuit state and function via the repair technician; SSCFI will be able to offer explanations of its diagnoses at various levels of detail; and SSCFI will provide active test assistance to the technicians in the field. In particular, SSCFI will provide technicians a uniform abstract interface to the underlying circuit test systems.

We are also exploring applying SSCFI's accumulated design-parsing knowledge to the problem of automatic design database cleanup. Database quality is a large and chronic problem, in spite of several costly efforts to address it. It is prohibitively costly to update the designs by hand, but automated methods -- cross-comparing data from multiple databases, combined with explicit testing for verification -- have great potential.

Conclusions

SSCFI was the first fully integrated, on-line operations support system at GTE to use knowledge-based technology. Many competent knowledge-based systems have failed to achieve user acceptance; SSCFI has been successful because it successfully captured the testers' expertise in a form that can be deployed *cost-effectively* throughout the organization, resulting in both substantial monetary savings and significant improvement in testing quality.

Acknowledgments

Many people have contributed significantly to this project. We wish to acknowledge the financial and organizational support of Clark Erskine, Bob Abe, Linda Whittaker, and Jim Crutchfield. We thank Gene Nordgren, John Hernandez, Tony Leto, William Mattil, Joe Dwyer, and Mike Warren for many spent hours patiently explaining to us the details of special circuits and testing. Hank Henning deserves especial thanks for his multiple contributions to SSCFI's acceptance as a legitimate and valuable tool: as California SSCFI administrator, National SSCFI Coordinator and advocate, and testing expert. Mike Sullivan, Brad Orner, Hansel Wan, and Mark Weissman contributed significantly to the design and implementation of SSCFI. We thank Rick Alterman for his comments on an earlier draft of this paper.

References

Acharya, A. 1994. Scaling up production systems: Issues, approaches, and targets. *Knowledge Engineering Review*, 9(1): 67-72.

Brightware Corp. 1988. ART-IM Programming Language Reference.

Clancey, William J. 1988. Acquiring, Representing, and Evaluating a Competence Model of Diagnostic Strategy. In Chi, M.; Glaser, R.; and Farr, M. editors, *The Nature of Expertise*, 343-418.

Davis, R. and Hamscher, W. C. 1992. Model-based reasoning: Troubleshooting. In Hamscher, W.; Console, L.; and De Kleer, J. editors, *Readings in Model-Based Diagnosis*, 3-24. Morgan Kaufmann, San Mateo CA.

Giarratano, J. and Riley, G. 1993. *Expert Systems: Principles and Programming*. PWS Publishing Company, Boston.

GTE Services Corporation 1990. GTE Technical Interface Reference Manual.

Libes, D. 1991. Expect: Scripts for controlling interactive processes. *Computing Systems*, 4(2), University of California Press, Berkeley, CA.

Osterhout, J. 1994. *Tcl and the Tk toolkit*. Addison-Wesley, Reading, MA.

Roth, Emilie M. and Woods, David D. 1989. Cognitive task analysis: An approach to knowledge acquisition for intelligent system design. In Guida, G. and Tasso, C. editors, *Topics in Expert System Design*, 233-264. Elsevier Science Publishers B.V.