NYNEX MAX: A Telephone Trouble Screening Expert

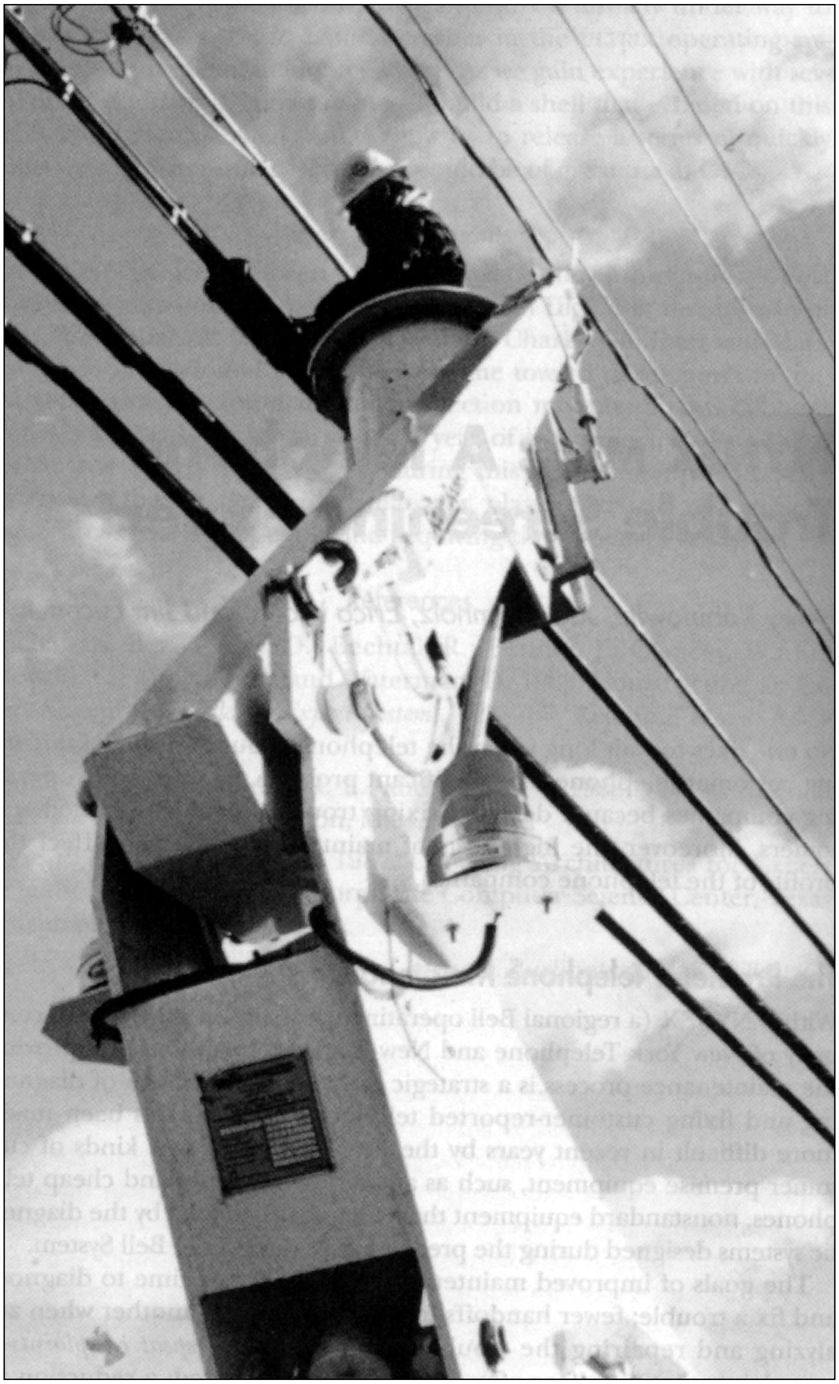
Henry Rabinowitz, Jack Flamholz, Erica Wolin, and Jim Euchner

No one likes to wait long when the telephone is out of order. Maintaining customer telephones is a significant problem for telephone operating companies because delays in fixing troubles mean dissatisfied customers. Moreover, the high costs of maintenance adversely affect the profits of the telephone companies.

The Problem: Telephone Maintenance

Within NYNEX (a regional Bell operating company and the parent company of New York Telephone and New England Telephone), improving the maintenance process is a strategic priority. The problem of diagnosing and fixing customer-reported telephone troubles has been made more difficult in recent years by the proliferation of new kinds of customer premise equipment, such as answering machines and cheap telephones, nonstandard equipment that was not anticipated by the diagnostic systems designed during the predivestiture days of the Bell System.

The goals of improved maintenance are: a shorter time to diagnose and fix a trouble; fewer handoffs from one person to another when analyzing and repairing the trouble; a reduction in *repeat complaints*—complaints that resurface after the trouble was cleared; a reduction in



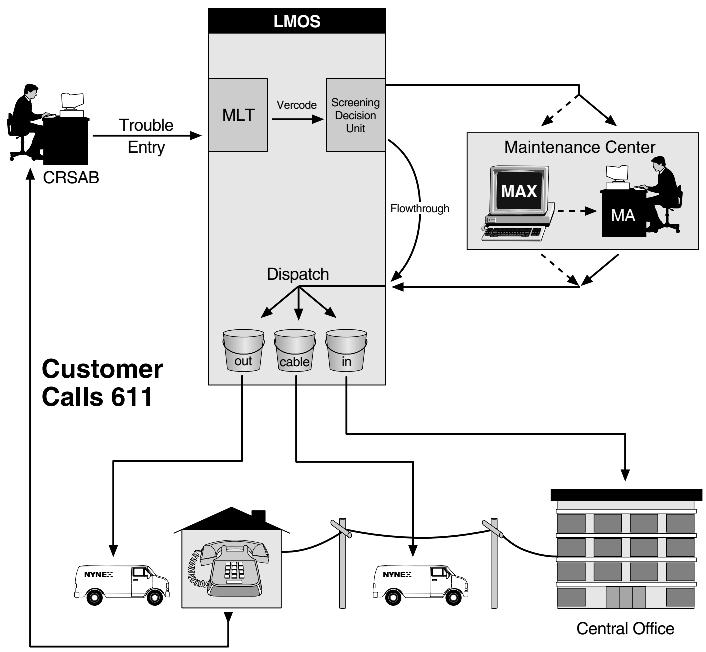


Figure 1. Trouble Flow with and without MAX.

false dispatches—the sending out of a repair technician when the problem is actually in the customer premise equipment, or there is no trouble found at all; and a reduction in *double dispatches*—the sending out of a repair technician to the home when the problem is in the cable, the central office, or some permutation of these locations.

Customer troubles are currently recorded by a Centralized Repair Service Answering Bureau (CRSAB) that answers calls to 611.[1] The troubles are then screened in a maintenance center where they are diagnosed by *maintenance administrators*, who then dispatch the troubles to technicians in the field or the central office. The maintenance administrator first diagnoses where the trouble is: the customer premise equipment, the customer's wiring, the cable facilities (underground or aerial), or the central office (switch, frame, or program control).[2] If the trouble is in customer premise equipment, the maintenance administrator can talk with the customer to help diagnose the trouble.

The entire operation runs on 1970s-style automation: A mainframe computer system, the loop maintenance operation system (LMOS), which was introduced by AT&T in the early 1970s, replaced an earlier operation based on paper slips and dispatch wheels. The CRSAB clerks enter the troubles into LMOS; the maintenance administrators receive the troubles from LMOS and dispatch them through LMOS to the field technicians; the technicians receive their assignments on hand-held LMOS terminals and enter the final status of the troubles back into LMOS.

LMOS has access to a mechanized loop test (MLT) facility. A clerk in CRSAB invokes MLT as soon as the customer identifies the telephone number in trouble. MLT creates an electrical profile of the wire pair, or *loop*, between the customer's telephone and the central office. MLT results are the primary source of information for diagnosing the trouble.

Opportunity for an Expert System for Telephone Trouble Diagnosis

To the Expert Systems Lab at NYNEX, the diagnosis of telephone troubles provided a clear opportunity for an expert system for the following reasons: First, some people are much better at analyzing troubles than others. Second, diagnosis involves the analysis of much information—the electrical test data, the type of switch equipment, and the distance of the trouble from the central office. Third, the diagnostic reasoning process must at times proceed with incomplete or inaccurate data. Fourth, new types of equipment are always being introduced in the network, requiring an evolution of the diagnostic rules.

Historically, diagnosis was performed by test desk technicians, or *testers*, whose training enabled them to perform electrical tests using specialized test equipment on a customer's line. Testers had a good understanding of the electrical principles of telephone operation. As part of the first generation of automation, the work of testers was largely replaced by the MLT facility, which automatically carries out the tests and reports their results to LMOS. The diagnosis could then be performed by maintenance administrators, who no longer needed as deep an understanding of the electrical basis of the telephone network, but knew how to read the MLT screen and apply its information in a formal way to the diagnosis of a customer trouble.

In a second stage of automation, LMOS Generic 3 included a *screening decision unit*, a primitive rule-based system for diagnosing troubles based on a condensed version of MLT data called the *vercode*, a two-character code intended to summarize the MLT results. Unfortunately, the vercode does not preserve enough information to allow the screening decision unit to make an optimal decision. Some locations rely on the screening decision unit to make dispatch decisions because their heavy load of troubles makes human screening difficult. Those locations that rely on the screening decision unit to diagnose their trouble load generally show a higher rate of double and false dispatches.

MAX was designed to emulate the work of a human maintenance administrator, that is, to use the MLT test results, together with other information such as the weather, to make a screening diagnosis. The only exception is that MAX would have the option of referring difficult troubles to a human maintenance administrator. A goal of MAX is to reduce the number of double and false dispatches.

Knowledge Acquisition

Part of the problem in building a knowledge-based system in an already automated field is that much of the knowledge in the domain is disappearing. Only seasoned veterans who once worked as testers have deep knowledge of the telephone system from the ground up. We turned to one such expert from New England Telephone, Ed Power, who was able to suggest rules that were more subtle than those used by many maintenance administrators today. For example, his rules are based on a three-point electrical test, but most maintenance administrators use data from the less reliable two-point test.

The series of interviews with the expert lasted several months. After modeling Power's diagnostic ideas in a knowledge base using the ART expert system shell, we compared the results of the rules on a set of troubles with Power's diagnosis. The comparison led to further refinements in the knowledge base. This process continued through several iterations.

External Architecture

MAX emulates a human user sitting at an LMOS terminal. MAX receives a trouble on an emulated LMOS terminal screen, obtains MLT data on another emulated terminal screen, makes an expert diagnosis, and enters the recommended dispatch instructions on the original LMOS screen. MAX's recommendations take the form of a status code, which directs the trouble to the correct dispatch pool, and a narrative in which MAX explains what it thinks is the cause of the trouble.

The advantage of emulating a human maintenance administrator's interface is that no changes in the host systems or the maintenance center's operations are necessary. Because MAX works on one trouble at a time, even if it fails, it cannot disrupt the operations of the maintenance center. The host system tracks each trouble; if a trouble assigned to a maintenance administrator or MAX times out, LMOS reassigns it to another maintenance administrator.

Because MAX emulates a human maintenance administrator, it is easy for management to track its performance. All the management tools for monitoring a maintenance administrator's performance can be used to monitor MAX's performance. For example, managers can monitor MAX's performance by checking the pending trouble queue for MAX's employee code or, retrospectively, by checking the results of troubles screened by MAX's employee code. Thus, the integration of the expert system can proceed with minimal change to the maintenance center's work flow. When people ask, as they always do of a new system, "Did MAX cause this problem?" we can always answer by saying, "Could a human maintenance administrator have done this?" MAX cannot do what a human cannot do. This design decision helped to smooth MAX's deployment. The ability of a workstation to emulate several terminals and run an expert process simultaneously makes this design possible.

The disadvantage of using terminal emulation as the interface to a host application is that the majority of the work in implementing the system is devoted to getting the communications right; the design of the knowledge base was by far the easiest part of the system. We are forced to use terminal emulation because the host system, LMOS, was purchased from another company, AT&T, and offers no application-to-application communications interface. Terminal emulation is complicated by the fact that screens change periodically, with each new release of the host application. In general, screens are not designed for computer "users" but for human users, who can more readily adjust to changes in the position and contents of a field. Unfortunately, at the moment, there is no alternative to terminal emulation. For an expert system to be of strategic significance, it must interact with host databases; to do so, it needs to use terminal emulation until host systems provide better interfaces.

To the host system, MAX looks like just another maintenance administrator with a distinct employee code. To cause LMOS to feed troubles to MAX, MAX executes an administrative LMOS command that establishes a queue of pending troubles for MAX's employee code and modifies the LMOS screening rules to send appropriate troubles to this queue.

Internal Architecture

Internally, MAX consists of an *expert agent* that communicates with LMOS through a *session manager* that handles multiple emulated terminals. MAX is implemented on general-purpose UNIX workstations. We chose UNIX workstations over Lisp machines because of their relatively low cost and because UNIX with windowing forms a good environment for software deployment and user interface design. We used Sun 3/60 and

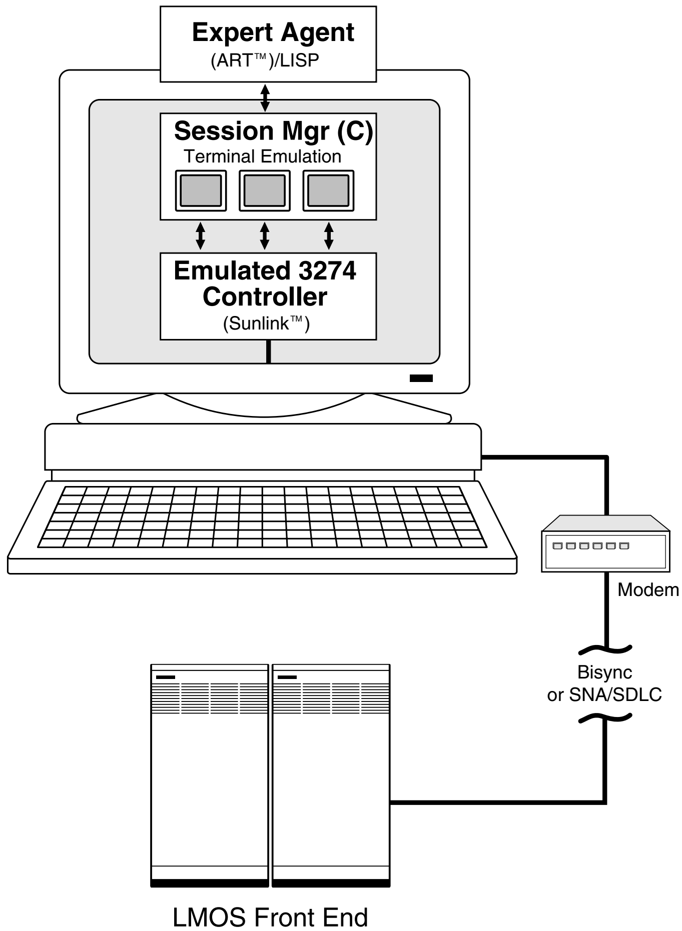


Figure 2. The MAX Internal Architecture.

Sun Sparcstation 1+ workstations to deploy MAX. The MAX expert agent's knowledge base is implemented in ART with additional Common Lisp code. The communications interface between LMOS and the MAX expert agent is handled by the session manager, an application-level protocol interface-building tool that was designed for use in MAX. The session manager has proven to be such a valuable tool for implementing host-workstation dialogues using multiple terminal emulation sessions that it has already been reused in four other applications. We used Sun's Sunview interface to build the user interface to MAX and FrameMaker to provide an online hypertext help facility.

ART-Lisp Expert Agent

The heart of MAX is the expert agent's knowledge base. The expert agent is written as a set of about 75 rules that follow the human expert's reasoning process. The MAX expert agent bases its diagnosis on five elements: First is the MLT electrical signature, including voltage, capacitive and longitudinal balances, and AC and DC resistances. MLT sometimes also provides the distance of the trouble from the customer and from the central office. There can be more than one invocation of MLT. Second is the type of switching equipment to which the customer's line is attached. (Each type of switch has a distinct electrical signature, and MLT can be fooled by certain signatures on certain switches, drawing the wrong conclusion from its own data.) Third is the class of the customer's service. Fourth is the weather. Different rules pertain in wet and dry weather. A human user must tell the computer what the weather is like outside. Fifth is the number of stages of cable facilities between the customer and the central office.

The expert agent reasons by forward chaining from the available trouble data. If more than one diagnosis is reached, an arbitration phase chooses the best diagnosis. The arbitration phase works as follows: First, select the diagnosis derived from the highest-authority rule. Rules are assigned authority as members of rule groups. The authority of a rule group is based on the type of information involved. For example, rules based on water have higher authority than rules based on the type of equipment. Second, if two diagnoses have the same authority, choose the diagnosis that is cheaper to implement. For example, it is cheaper to dispatch a technician to the central office than to dispatch to the customer's home. The rules are designed conservatively; if there is insufficient information to choose a firm diagnosis, MAX passes the problem to human maintenance administrators for further diagnosis or testing.

Under certain conditions, MAX can decide that the original data from MLT, performed when the trouble was called in, are inconclusive. The expert agent can request a fresh MLT. It can then base its diagnosis on the combined information from the old test and the fresh test. To request a fresh test, the expert agent sends a request to the session manager, which sends a request to LMOS by typing on the emulated terminal.

The MAX expert agent's diagnosis is expressed in a canned narrative that is used by human screeners and repair technicians who subsequently receive the trouble. It might say, for example, "check for open in drop wire" or "needs talk test; possible CPE trouble." The diagnosis is also expressed in a status code that tells LMOS where to send the trouble. The trouble can be sent to a human maintenance administrator for further testing, a cable technician, a technician who will go to the customer premises, or a technician in the central office.

MAX's rules can be customized to local conditions by a set of parameters, which the individual maintenance centers can change and tune over time. In designing the rules, our philosophy about parameters was that they should be as few as possible, they should enable all sites to use the same set of rules, and they should each have a physical meaning. These goals proved difficult to attain; we began with 8 parameters and, after a second release, ended with 29. However, the enhanced set of parameters allows the MAX administrator to more closely tailor MAX's operations to local conditions.

Session Manager for Terminal Emulation

The session manager is a tool for writing application-level protocols between various hosts and processes. In MAX, we used the session manager to communicate between the expert agent process and three emulated terminal sessions associated with Bisync LMOS terminals. The Bisync communication is implemented using Sun's Sunlink package, which allows a Sun to emulate a 3270 controller. We also ported MAX to a VTAM-SDLC communications environment using Sunlink SNA.

The session manager is a data-driven program. It allows the programmer to specify a set of agents, each of which controls a dialogue between the application and a host application or other external process. Each agent is specified as a set of messages to be recognized and a finite-state machine that controls its actions. Each agent has a device, which controls its communication with the external host by means of, for example, terminal emulation, printer emulation, or UNIX pipe.

For each agent with a terminal emulation device, its messages are specified as a set of screens, or *masks.* The masks can be specified using fields, which are regular expressions anchored at particular row and column positions. A mask can also be specified as a Boolean combination of fields and other masks. A mask can be relocated to different parts of the screen so that a complex set of masks can be constructed from common components.

Each agent has two mailboxes: an external mailbox that receives events corresponding to incoming masks on the agent's device and an internal mailbox for receiving messages from other agents, allowing the agents to cooperate and coordinate their work.

In the original implementation of the session manager, the agent specifications were written in Lisp-style S-expressions. A Lisp preprocessor converts these specifications into object-oriented C code, which is then compiled with the generic session manager code.

The session manager also contains a timer facility that allows an agent to send a message to another agent (or itself) after a specified time interval. This facility is useful in emulating a maintenance administrator's ability to periodically hit a key to request a new trouble.

Because of the many users of the session manager, we reimplemented it in C++, using LEX and YACC for preprocessing the agent specifications. The rewrite was necessary to thoroughly disentangle MAX from the session manager. We successfully used the session manager in four other applications, using Bisync, SDLC, and asynchronous terminals as well as UNIX pipes.

The session manager offers a number of useful tools for debugging, including a log file from which the entire session can be replayed. The log file created by the session manager contains all input events from external sources and all timer events. In playback mode, the session manager can replay the events from the log file, possibly choosing some sources from live input and some from playback input. This approach enables the programmer to recreate communications problems and analyze them in the lab, where a greater number of debugging tools are available. We have also played back log files on site and over the phone for rapid troubleshooting.

Current Deployment and Benefits of MAX

MAX is currently running in 42 maintenance centers in New York Telephone and New England Telephone (virtually every residence-oriented maintenance center). It screens 38 percent of all troubles, that is, about 10,400 troubles every day.[3] Because each maintenance center can control how many troubles flow through MAX, some centers make greater use of MAX than others. Some centers have over 50 percent of their troubles flowing through MAX.

Part of the variation in the percentage of troubles screened by MAX is because MAX currently handles only residential and small business troubles. The variety of customer premise equipment in large businesses makes diagnosing their troubles more complex. A version of MAX for large businesses is under development.

Another reason why MAX does not handle all troubles is that LMOS can correctly screen certain troubles on its own, using its screening decision unit. We did not need to replicate this capability. Also, troubles with certain "handle codes" assigned in CRSAB bypass the screening decision unit and, hence, MAX. For example, when a troubled line is identified as a component of a damaged cable, it is automatically attached to the cable failure ticket for this damaged cable, without passing through MAX.

Locations that formerly relied heavily on the LMOS screening decision unit, primarily in New York Telephone, find that MAX's greatest benefit is in reducing false dispatches in the field. Locations that formerly relied heavily on human screening of troubles, primarily in New England Telephone, find that MAX's greatest benefit is in reducing the number of maintenance administrator hours needed for screening troubles in the maintenance centers.

As with other expert systems, the measurement of MAX's benefits is not a simple matter. We performed studies with small samples and highly accurate data and studies with large samples but somewhat noisy data. With few exceptions, MAX's diagnostic accuracy was universally accepted. In each location where MAX was deployed, employees would challenge MAX's more unusual diagnoses and were often pleased to find that MAX reasoned correctly, if unconventionally, in making its conclusions. A comparison of MAX's diagnoses with those of human experts found 96-percent agreement.

One way to measure the benefits from MAX is to review a set of troubles diagnosed by MAX and retrospectively examine their final dispositions. One such study was performed on a set of troubles randomly selected from a base of 5158 MAX-handled troubles in 4 maintenance centers—2 in New York, 2 in New England—over a 1-week period. Such a study required careful analysis of each trouble by an expert, precluding a large sample of troubles. The final dispositions of MAX-diagnosed troubles were contrasted with the way these same troubles would have been diagnosed by the LMOS screening decision unit before MAX was installed. The purpose of the study was to see how much human testing (maintenance administrator work in the maintenance centers) and how many unnecessary dispatches (technicians in the field) were saved by MAX. The results at the four maintenance centers are shown in table 1.

Maintenance centers 2 and 3 are in New England, and 1 and 4 are in New York. Note that in maintenance center 4, MAX actually increased the amount of maintenance administrator testing. Even a small-percentage savings in the number of dispatches translates to significant dollar savings because each dispatch typically involves at least one hour of time by a highly trained craft worker.

The aggregate results for these tests are as follows:

Number of troubles examined:	593
Number of tests saved:	23
Number of dispatches saved:	25

Similar studies were performed in other maintenance centers with equally promising results.

Center	Tests Saved (%)	Dispatches Saved (%)
1	9.1	5.3
2	3.8	0.8
3	5.7	4.2
4	-3.7	10.6

Table 1. Test Results Comparing MAX and Human Testers.

However, the accuracy of individual decisions is not the only way to measure MAX's benefit. Another way is to observe changes in the false dispatch rate in the maintenance center as a whole. Because MAX sometimes passes its diagnosis of the trouble to human maintenance administrators for further screening, it is not sufficient to view MAX's performance in isolation from the maintenance center.

A large-scale study was performed involving nine maintenance centers in New England. With a database query system associated with LMOS, data were pulled for two seven-week periods, one from a pre-MAX baseline and one from a period exactly one year later with MAX in place. The one-year-later time period was selected so that weather and other seasonal factors would have minimal impact.

The results are as follows:

	Troubles Screened	False Dispatches
Baseline	177516	16663 (9.39%)
With MAX	178832	15669 (8.76%)

The drop in false dispatches from 9.39 percent to 8.76 percent constitutes a 6.7 percent improvement and represents nearly a thousand saved dispatches in 9 maintenance centers over the 7-week period. To these savings must be added the savings in maintenance administrator testing, which in New England locations is significant.

These studies, combined with other similar studies, conservatively place the savings in maintenance center operations at several million dollars each year—at a minimum.

Implementation History and Effects on the Maintenance Centers

The initial design of MAX began in December 1986. The knowledge base was first tested against a set of troubles in mid-1987. The first live test of MAX in a maintenance center occurred in Manchester, New

Hampshire, in February 1989. MAX was programmed by 3 people, taking 2-½ years from inception to first deployment. Then, fate intervened in a curious way. Just before the scheduled deployment of the first production version of MAX, NYNEX underwent a strike by its unionized employees. MAX was drafted to help the management staff run the maintenance centers without their normal complement of craft workers.

We had not intended to deploy rapidly. In fact, we had developed software and a strategy for prospective preinstallation assessment of MAX site by site. We had planned a careful, slow rollout of MAX, with time to evaluate its performance in each maintenance center, carefully tune its parameters to local conditions, train local users, and build the trust of management and workers before moving on to the next site. Under the circumstances, these steps had to be deferred.

Because of the strike, MAX was deployed in 23 maintenance centers within 5 months. The strike conditions enabled a rapid deployment because normal operating procedures were temporarily bypassed. The rapid rollout of MAX enabled the maintenance centers to cope with the volume of troubles encountered during the strike. The users of MAX found it was especially helpful in identifying customer premise equipment troubles that could be handled by talking to the customer, thus saving dispatches to the field.

Once the strike was over in December 1989, we began to revisit the MAX installations. The most striking observation we made was that certain features of the MAX system, designed and tested in New England, did not work appropriately in New York City. New England differs from New York in its weather, the age of its plant, its operating procedures and regulatory rules, and its trouble load.

New York City is exceptional in many ways, including its telephone network. For example, MAX correctly identified most of the problems in many areas of New York City as lying in the cable facilities. However, this diagnostic fact, although correct, was not useful to the maintenance centers, which lack the personnel to correct all the cable problems in the city.

In one particularly tense and memorable meeting, the management of three New York City maintenance centers told us that if we didn't reduce MAX's propensity for cable dispatches, they would shut it down indefinitely. They had grown accustomed to dealing with the chronic shortage of cable splicers by *swapping pairs*, a procedure that quickly restores service but defers cable maintenance. MAX was actually creating extra work for them, forcing them to manually restatus a large portion of its cable dispatches as swaps.

We found ourselves faced with a dilemma: Either retain the purity of the knowledge base and face shutdown in the centers, or make MAX an automated accomplice to a practice that senior management did not want to encourage. Our solution was a compromise. We agreed that a subclass of the cable troubles, namely, cross-battery troubles, would always be marked for cable dispatch, with a new voltage-threshold parameter provided to control MAX's sensitivity to cross-batteries.[4] The remaining cable troubles would default to cable dispatch, but the maintenance centers could restatus them for pair swaps inside MAX by setting a parameter. In this fashion, we supported the maintenance centers' need for an established "workaround" without compromising the knowledge base.

Regulatory differences between New England and New York created another crisis. In New York State, troubled lines marked "out of service" must be fixed within a certain time interval, or the customer is entitled to a rebate. The out-of-service indication is piggybacked onto a three-digit code on the trouble ticket that MAX receives, a fact that the New England expert was unaware of. MAX was unwittingly overwriting the out-of-service indication in certain New York troubles. As a result, some New York centers were charged with undeserved rebates and shut MAX down for a few weeks. We addressed this problem by introducing a parameter to allow local control.

Our solution to most regional differences was to enlarge the set of parameters that allows each maintenance center to tune MAX to its local needs without changing the knowledge base itself. Thus, one knowledge base can still suffice for all our installations.

Maintenance of MAX

The Expert Systems Lab at NYNEX Science and Technology found that maintenance of MAX is an issue that cannot be ignored. Terminal emulation means that MAX is vulnerable to changes in corporate networking policies and host application screens. New kinds of customer equipment or switch equipment and changes to operating procedures require changes in MAX. The fact that it is embedded in the day-to-day operations and communications of the company makes MAX more vulnerable to change than management information system–type applications—the knowledge in knowledge-based systems ages quickly. Moreover, the Lisp and ART languages involved in MAX make it difficult to hand off to a management information system organization. Therefore, the Expert Systems Lab chose to assume the responsibility of maintaining MAX, with the idea that this process will help us channel MAX's evolution in a direction desirable to the users. We brought some people from the operation side of the telephone companies into our lab to help us support the product. Their presence not only helps us support the existing product but also anticipate the future needs of the users in the maintenance centers.

A second release of MAX in March 1990 greatly increased the number of parameters that are controllable by the individual maintenance centers. Increasing the number of parameters allows MAX to more flexibly meet local needs without requiring changes in the knowledge base. The problem with introducing more parameters is that eventually a second expert system might be needed to help set the parameters! In reality, we followed the release by having a human expert visit each maintenance center to tune its parameters based on local needs. The increase in the number of parameters can point to gaps in the knowledge base, which, if filled, would reduce the parameter set to a manageable size.

Reactions to MAX

Skepticism of MAX's abilities accompanied its arrival in each maintenance center and was generally dispelled by the examination of a few of MAX's diagnoses. The opposite problem appears over time: People begin to trust MAX too readily. Its status as a computer lends it undue authority; the users assume that MAX makes the right decisions without carefully evaluating MAX's performance. Changes in the operating environment that would affect MAX's rules are not always entered into MAX. MAX requires continued vigilance by the system administrator and local management. Some centers continue to question MAX's value and have configured it to process few customer troubles, but they are a small minority.

Workplace Issues

In general, management is drawn to expert system technology by its promise of increased productivity, but clerical and craft workers see it as having a destabilizing effect on their work environment. Management might find it easier to initiate change in the workplace by introducing obedient, pliable computer systems than by changing policies, operating procedures, and contracts.

Although NYNEX does not traditionally lay off employees, MAX will lead to some reassignment of employees. Many maintenance administrators are concerned that MAX will be used to reduce the number of maintenance administrator positions. They understandably see MAX as computerized competition—MAX can diagnose a trouble in 10 seconds, but a human maintenance administrator takes about 3 minutes.[5]

Automation does not always reduce workload and jobs despite our impression to the contrary. For example, MAX is conservative in its dispatching strategy, leaving certain kinds of troubles for human analysis. In table 1, note that MAX increased the test load in maintenance center 4, which meant more work for the maintenance administrators. However, MAX also cut the dispatch rate, thereby enabling maintenance center 4 to reallocate outside craft workers from ongoing to preventive maintenance. The center subsequently experienced a decrease in new troubles. Although it was MAX that initially reduced the load on the outside craft workers, the consequent drop in report rate resulted from a decision by local management on the best way to reallocate these people.

Another example of the complex outcome of introducing MAX into the workplace is in the cable dispatch area. The fact that MAX identifies many problems in the cable facilities lends greater weight to these problems, even though the problems were well known beforehand. Thus, paradoxically, MAX might lead to creating more jobs in some areas (cable splicing) and reducing jobs in others.

The choice of how to reallocate maintenance administrators and field technicians after MAX's deployment is a significant management decision. The presence of MAX can allow maintenance administrators to move to positions where they can focus on more preventive maintenance work. However, the pressure to reduce costs in the maintenance centers might push managers to see the presence of MAX as an opportunity to reduce the number of maintenance administrators.

Centralize or Decentralize?

MAX is designed as a locally based workstation tool. Currently, the local users interact with MAX primarily by turning it on and off, announcing the weather to it, changing local parameter information, and checking its performance during the day. The developers of MAX would like to use its presence in the maintenance centers to provide more interactive, cooperative tools for analysis and training. Management and data processing operations staff are understandably concerned about having 42 UNIX workstations performing critical tasks at widely dispersed geographic locations that lack local system expertise. They prefer to centralize all the MAX machines.

Those who favor remaining decentralized argue that the company should pay the higher administrative cost of distributed workstation maintenance to realize much greater cost savings in customer loop maintenance. Given that MAX already takes a significant amount of work and, with it, some control away from the centers, the proponents of remaining decentralized question the impact of taking away the machine itself.

The proponents of centralization counter that loop maintenance is a highly reactive, pressured job that leaves little time for analysis. Thus, no matter how powerful and sophisticated the workstation tools are, the benefits are limited by the job and the environment. They argue that one might as well at least enjoy the benefit of centralized hardware maintenance. The centralization-decentralization debate has not been resolved.

Future Enhancements

An enhancement that is under way will make the MAX knowledge base available as a server to other client applications. For example, when a new telephone is installed, the line is tested on the morning before the installer goes out. The installation process would benefit from using MAX's diagnostic abilities to analyze the test results. Similarly, if the operator in CRSAB had MAX's expertise available, s/he could possibly close out certain troubles during the initial contact with the customer. Moving MAX's expertise into the initial customer contact can transform the work environment by reducing the number of handoffs of the trouble. Both these applications are actively being pursued. In each case, MAX's knowledge base would have to be restructured to separate MAX's knowledge from the policy requirements of MAX's current job as a trouble screener.

We are actively investigating an adaptive MAX that uses machine-learning techniques to improve performance. The greatest problem in this effort is obtaining a reliable feedback mechanism; the data on final disposition of closed troubles can be noisy. We are examining inductive, neural net, and explanation-based learning approaches.

A further area for future enhancements is giving maintenance administrators more tools to examine MAX's performance. A maintenance administrator should be able to inquire how MAX handled a particular trouble on a particular date and ask MAX why it made this diagnostic decision. This inquiry feature would be useful both as an educational tool and as a means of improving user feedback. Such a tool would make MAX friendlier to human maintenance administrators and might improve their perception of it.

We would like MAX to be able to read the free-form comments entered by the clerks in the original interaction with the customer. They use common codes for complaints, such as "HOOL" (hears others online). These codes could be incorporated into the evidence that MAX can sift for diagnosis.

One future enhancement that has often been requested is to connect MAX to more host systems so that it can incorporate more information in its decisions. For example, if MAX connects to the billing system, it can determine whether the customer has lost service because the bill was not paid.

Conclusion

MAX demonstrates the practicality of using an expert system to diagnose telephone troubles. MAX is now a mature and widely deployed expert system whose integration in the NYNEX telephone companies' maintenance operation has been smoothed by its emulation of the human maintenance administrator's interfaces. The deployment of MAX has shown the value of a workstation-based expert that can talk to multiple host screens through terminal emulation. Although MAX emulates a human maintenance administrator's computer interfaces, it uses a knowledge base with a deeper knowledge of the electric basis of telephone diagnosis than is possessed by many human maintenance administrators. MAX's design included reusable components that have since proven their value in other expert systems as well.

Notes

1. In some areas, the Centralized Repair Service Answering Bureau has a different telephone number.

2. Troubles lying between the central office and the drop wire to the customer premises are said to be in the cable facilities. Cable can be aerial or underground.

3. This estimate is for all NYNEX locations as of October 1990. New York Telephone alone sends 29 percent of its troubles through MAX, about 6,100 troubles each day. The percentage is based on the total number of customer-reported troubles for residential and small business telephones.

4. Cross-battery troubles are caused by an undesired completion of the customer's circuit with a foreign battery, often indicating trouble in the cable. MAX's sensitivity to these types of troubles can be controlled by setting the threshold of the voltage at which such a problem is noted.

5. MAX can take as long as two minutes to diagnose a trouble if MAX needs to ask for a retest of the telephone line. Most of this time is spent just waiting, so that the retest does not immediately follow the initial test.