OPERA: A Highly Interactive Expert System for Outside Plant Engineering

Gary Lazarus, Liên Tran and Marty Baade

Expert Systems Laboratory
NYNEX Science & Technology
500 Westchester Avenue
White Plains, NY 10604
lazarus@nynexst.com

Abstract

OPERA is an expert system developed at the Expert Systems Laboratory at NYNEX Science & Technology to assist outside plant telecommunications engineers in the planning and implementation of sophisticated electronic equipment. The engineer communicates with the system via intelligent documents that are schematic in their design and dynamic in their operation to facilitate the conveyance of necessary and sufficient information. Expertise from the top experts in the company was culled and built into the system to create an expert system that surpasses the engineering knowledge of any single engineer in the company. The domain and project objectives are discussed, the system architecture and development, testing, deployment and maintenance issues are presented and the use of artificial intelligence techniques is justified. A benefits analysis supports the positive contribution of the project.

Outside Plant Engineering

Outside plant engineering in the telephone companies is a dynamic, technologically sophisticated domain. The complexity of the electronic equipment and the sheer size of the task of providing dial tone in an expeditious manner to anyone who wants it conspire to make it a domain that defies conventional software solutions. This is plainly evident in the paucity of systems installed to date. Yet, the magnitude and importance of the task demand that solutions be provided to maintain the high quality that has historically been characteristic of the telephone networks, and to maintain market share in an increasingly competitive business arena. Within NYNEX (a regional Bell operating company and the parent company of New York Telephone and New England Telephone), improving the outside plant engineering process is a strategic priority. To meet these needs, the Expert Systems Laboratory at NYNEX Science & Technology Research and Development designed and developed OPERA, a knowledge-based expert system that employs numerous artificial intelligence techniques. Rule-based reasoning, dynamic forms-based user interaction and sophisticated successive-refinement knowledge acquisition techniques are used to produce a flexible, easy-to-use system that contains more engineering expertise than any single engineer in the telephone companies.

As is the case in most high technology fields, outside plant engineers routinely see new products, or refinements and enhancements to existing ones. The underlying first order principles of engineering are complex and not necessarily fully understood by all engineers charged with the responsibility of turning up working systems. Engineers who immerse themselves in the latest technology can perform very well. But, the effort required to maintain that ability is significant. OPERA (***O***utside ***P***lant ***E***ngineering and ***R***esource ***A***dministration), therefore, must provide a reasoning mechanism that emulates the best engineers and be flexible enough, within the realm of that mechanism, to be able to adapt to new and different solutions. The electronic equipment can often supersede or obviate current solutions and complicate existing database designs. Yet, humans have shown a remarkable ability to absorb the new information and recommendations and apply the equipment in innovative and practical ways. Our task, then, was to emulate not just the decision process, but its creation and maintenance as well. It became apparent to us that the engineers had a fairly intricate decision tree of rules by which they engineered systems. The best engineers have richer, more detailed and more accurate knowledge, as is manifested by their superior designs.

OPERA focuses on two areas of outside plant engineering: Pathways, or T1 carrier, and Digital Loop Carrier (DLC). T1 is a designation for a high capacity digital transmission of telephone signals within the local loop, or the wires that connect a central office with subscribers. Employing time division multiplexing, it operates as a 1.544 megabits per second pulse stream. Low in cost and high in reliability, it provides capacity for 24 voice grade telephone lines for each T1 line and can operate on copper or fiber optic lines. Intended for short-haul transmission, it requires digital-to-analog converters at the end points to permit interconnection with analog switching and transmission facilities.

Digital loop carrier also uses time division multiplexing. Using T1 lines as input, it translates the digital signals to analog voice grade lines provided to the customer. DLC essentially substitutes electronics for cable in the local loop. Electronic equipment exists on both ends of the T1, in either the digital central office switch or a separate central office terminal, and in the field at the remote terminal. The remote terminal equipment can be located in an office building basement, buried underground in a controlled environment vault or inconspicuously by the side of a road in a hut or cabinet. Because it substantially reduces the amount of outside plant, DLC is often referred to as pair-gain for the extra copper pairs it gains, or route relief for the reduction in feeder congestion it provides. Optionally, the signal can be concentrated, providing even further gain. This is accomplished by sharing facilities in an environment where a significant portion of the resources are idle at a given time. In addition to providing economic advantages, the digital loop carrier systems also provide a wide range of customer services, including single or multi-party channels, trunk or coin services and numerous special services such as off-premise extensions, digital data lines and alarms. Customer channels can be remotely monitored and tested and troubles can be easily isolated.

Expertise is required in the selection, configuration and placement of the carriers, multiplexers, DLC equipment and related electronic equipment and the housing for the equipment. The engineer must understand the expected demand of the area and provision the equipment to be able to meet that demand. Multiple vendors exist, offering not only a varied line of equipment, but also a large number of configurations of their equipment to meet all needs. Knowledge of and experience with specific devices is necessary and, at times, in short supply. Errors in judgment can be costly and disruptive to the operation of the network. Customers are becoming increasingly reliant on their telecommunications abilities to perform their daily tasks and, as a result, are becoming less tolerant of anything short of perfect service.

T1 orders are customer initiated and must be completed according to a very strict deadline. DLC jobs are initiated from within the engineering office by a planning engineer who determines that facilities are needed in a certain area and that area will be best served by a carrier system. Implementation engineers then do a site survey, determine if facilities already exist, and choose the proper DLC devices and plug-in components to provide the quantity and type of service projected for the area. Order forms are completed and sent to the group that purchases components. Then detailed work plans are issued to service technicians who install the equipment. The engineer is also responsible for obtaining right-of-way clearances, usually with the help of a right-of-way engineer, and for ordering alarm and test circuits for the equipment. Finally, the implementation engineer visits the installation and turns up the equipment, placing it into service.

The engineering process is enacted almost entirely through paper documents. Letters to the outside world, diagrams depicting equipment placement and configuration, order forms and planning sheets are all paper based. Prior to OPERA, these documents were mostly handwritten. Time consuming and error prone, these are the vehicles through which outside plant systems are designed and implemented.

Application Objectives

Completing forms that often require redundant or unnecessary data is very time consuming. It became obvious that mechanizing data collection and allowing electronic representation of forms could help to contain costs by reducing the time needed to design a system. Reducing the time to complete a job was of further benefit to the company by improving the marketability of the product and increasing the company's market share, which was being strongly contested. Importantly, automation also makes the job less tedious and vastly improves the spirit of the engineers. But, we also recognized the need for adding an expert component to the system, to improve the decision making and design process that precedes the actual implementation of a system. Not only would errors resulting from inaccurately completed forms be reduced, but errors in design would be avoided. The knowledge of the company's best engineers could be leveraged throughout all of the engineering offices. Costly redesign work could be brought to a minimum. Systems would be engineered correctly the first time and in a cost effective manner.

Training new engineers and advancing the knowledge of existing engineers would be greatly improved, faster and less costly. Methods and practices would be more easily rolled out and maintained through working software and help screens than by producing reams of written recommendations. The engineers would have at their disposal a state of the art computer system that would raise their technological awareness and provide a platform on which to build subsequent software solutions. A database of outside plant information would be available for future system.

At the Science and Technology Center we also saw an opportunity to place a highly interactive expert system into operation in nearly fifty locations. To our knowledge, this would be among the very first operational systems of its kind, and it would provide us with an opportunity to both advance our skills and knowledge of deploying leading edge systems and to help advance the state of the art of expert systems design.

Why AI?

Often times, the work that an engineer needs to accomplish is based on uncertain, incomplete or incorrect information. Records are often missing or inaccurate. Time constraints are severe, thereby not allowing proper investigation of the details of a job. And, of course, new engineers, who can require up to six months of training to attain a level where they can be expected to work on their own, are not always aware of the location, or even the existence, of much of the necessary information. Even when information is available, it is often in a form that is very difficult to obtain. Yet, the best engineers manage to overcome these obstacles and still produce a working system. Emulating their reasoning and remaining flexible enough to adhere to frequently changing guidelines became fundamental elements of our design, and seemed to be best addressed by a knowledge-based expert system approach. The application is not one that lends itself to algorithmic solutions. Instead, it is only solved through careful knowledge engineering, identifying scenarios and their solutions, and building a rule set to adequately address the domain. The ability to display the line of reasoning exhibited by the sequence of rule firings was important to our objective of training the engineers. An intelligent interface removes the burden of learning how to communicate with the system and lets the engineer concentrate on the task at hand.

Solutions that contained hard-coded models of equipment were doomed because of the long lead time required to keep the models current. Simple, forms-based packages lacked the intelligence to assist and teach the engineer, and required as much effort and learning as the current paper-oriented approach. They were adequate at reducing the paper overload, but involved significant time and money to keep current or to add new forms and features. Finally, any system that was not forms-based would require a steeper learning curve for the users and would necessitate that the entire outside plant engineering function be addressed before its introduction since it would remove the engineer from the current methods and practices and immerse the engineer in a new environment. The OPERA intelligent forms-based approach allows us to pick pieces of the domain and introduce them as they become available. The output, printed documents, is largely the same as that which was otherwise produced by hand.

OPERA

Since our goal was to provide a solution to the engineers as quickly as possible and we wished to keep the training effort to a minimum, OPERA maintains the concept of communicating via documents by displaying forms on a workstation screen. It is easy enough for a novice to use, yet sophisticated enough to be of value to an expert engineer. Help is available at the touch of a mouse button at all times. A detailed user's guide with sample jobs was available by the time the system was in beta testing. An information line is present at all times on the bottom of the screen to explain the functions of the three mouse buttons. The system can save and load jobs and copy and delete them as well. Rules provide detailed explanations of how values are derived and allow the engineer to override them when necessary. OPERA is designed to be an intelligent engineering assistant.

System Architecture

OPERA comprises four major components: a user interface, an inference engine, a database and an executive to oversee and control the operations. The system is built around documents, which are stored as text files of commands, written in a simple, easy to understand definition language that we developed. Documents are displayed in actual size using half of the OPERA screen. Mouse-sensitive areas on the documents represent variables whose values are asserted by the user via keyboard entry or menu selection, the inference engine or the database. Variable properties are defined in another text file, also using a simple to understand definition language that we developed. As values are entered by the user on the documents, they are passed to the inference engine, which searches the rule base and fires any matching rules. The process continues until all inferred values are processed and all matching rules fired. The results are then passed back to the user interface for display on the documents. The database serves to store and load data for jobs and to maintain information about equipment and locations. A print feature allows the engineer to print all or some of the documents for the job.

OPERA's executive module is written in C. The inferencing is handled by OPS-83, a production system developed by Production Systems Technologies. The user interface is written in Motif and the database, which resides transparently behind OPERA, is built using Oracle. The selection of off-the-shelf products was carefully considered for portability to numerous platforms, vendor support, reliability, performance and availability of experienced developers. We are currently running OPERA on a Sun SPARCstation 2. Text files exist for the configuration of the variables, the activation of the rules and documents individually and by class and for controlling logging, printing and other aspects of the system. All of these features can be controlled via a telephone link, facilitating the systems' customization and ultimate usability.

User Interface

The user interface is fast and easy to use. Displayed items include lines, boxes, text and variables and are specified in the document definition files. To display a document, OPERA opens the appropriate file and reads each command line, which specifies the item's type and location. Items can have conditions, making them dynamic and responsive to the current scenario. For example, when the system or user determines that a digital loop carrier must be placed in the central office the planning document will pop up text and fields relevant to the specific equipment selected. Otherwise, the information will not be displayed. This assures that the documents only contain information that is both necessary and sufficient to the application at hand. Documents can also have multiple pages. The system automatically advances to the appropriate next page and can change the

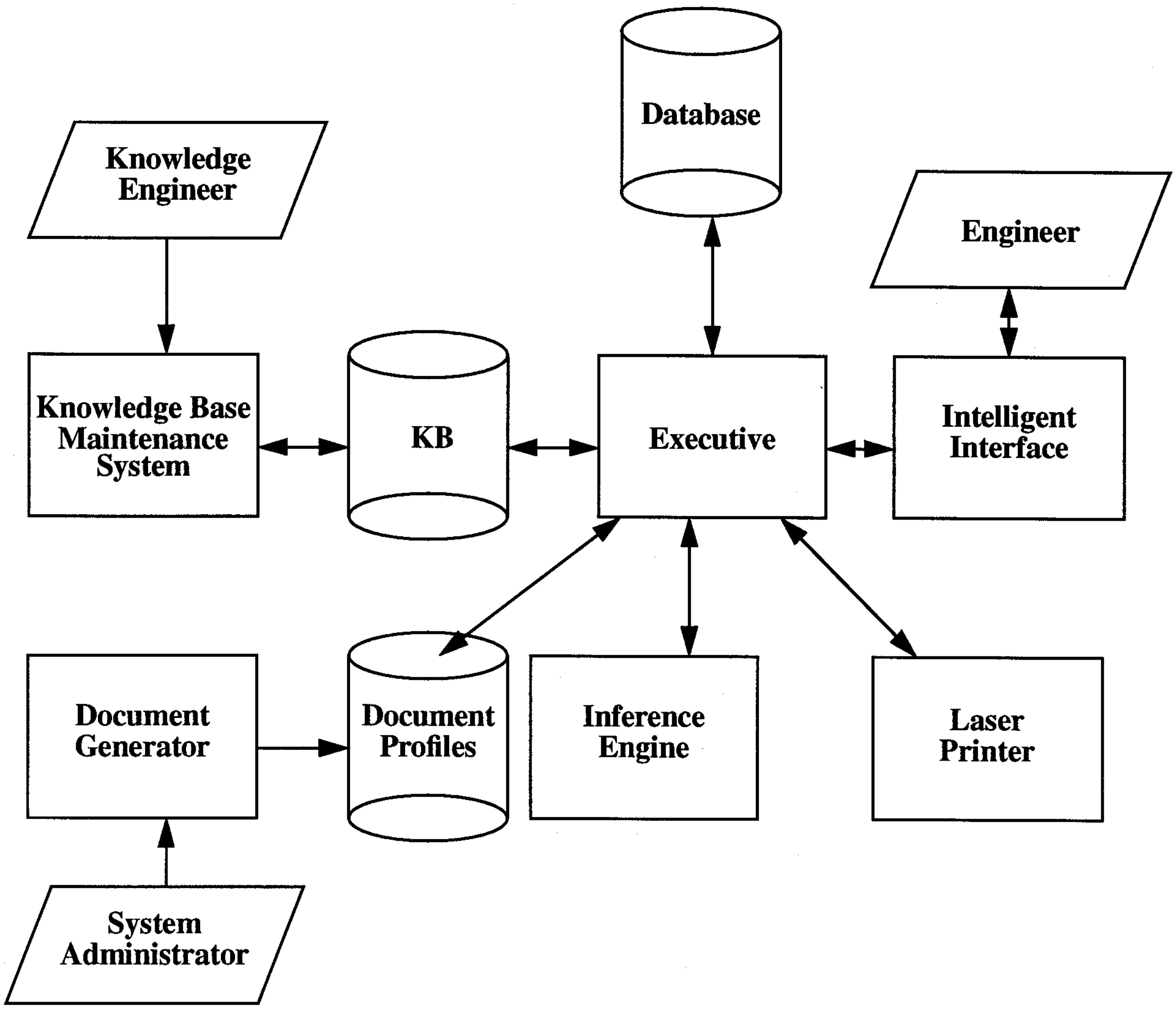

Figure 1: The OPERA Software Architecture.

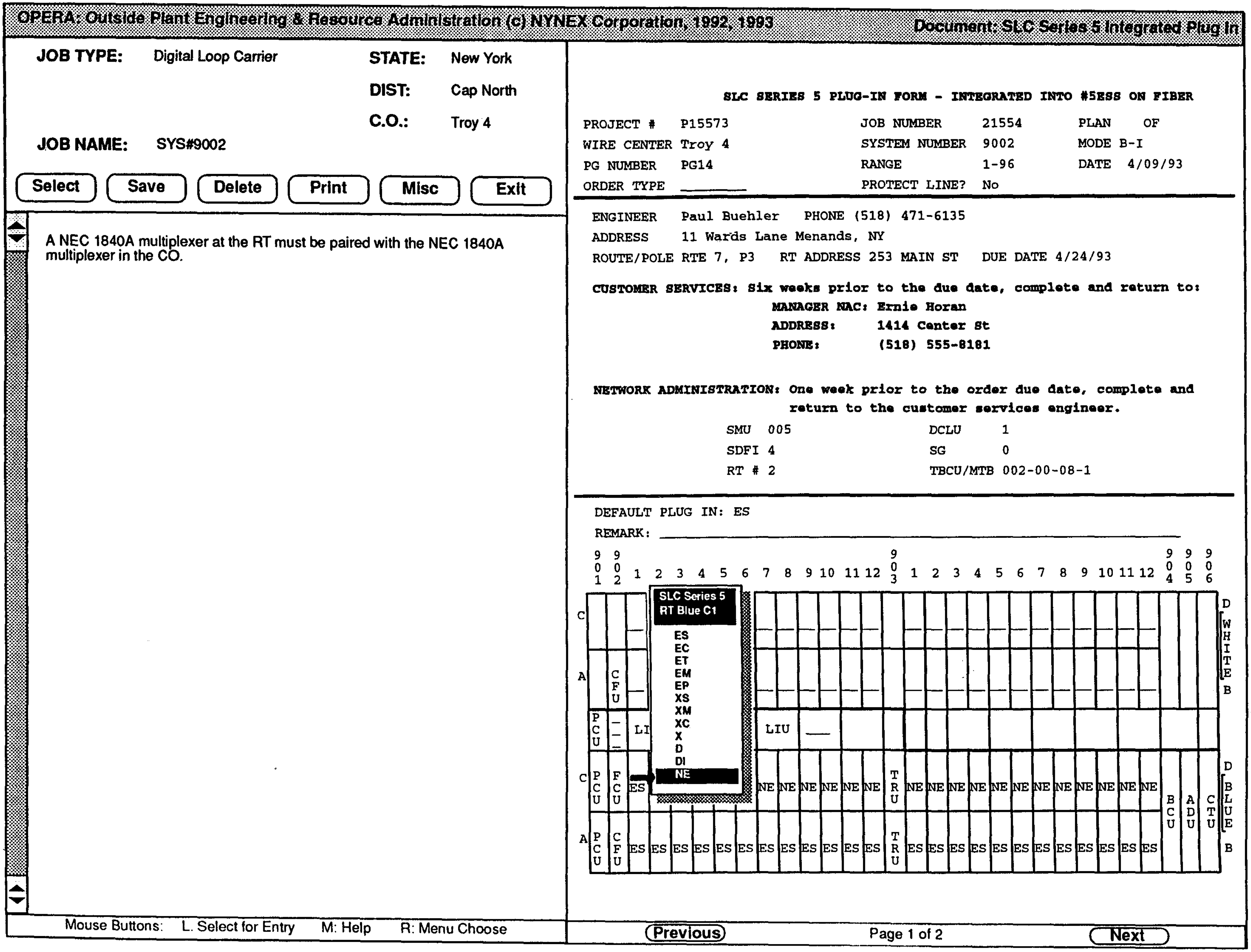

Figure 2: Using OPERA to Engineer a SLC Series 5 Digital Loop Carrier System.

total number of pages in a document dynamically, as well. If repeater apparatus cases are needed for a job, for example, a form for each case will be added to the planning document to allow for their design.

Variables are defined in a text file called the configuration file. Relevant properties of variables that are specified include help messages, corresponding working memory elements for use by the inferencing mechanism, data classes and types and a host of other system features. There are currently about 3500 variables in the system appearing on over 100 documents. Associated with each variable is a help screen containing a description of the item, an example of a value it can assume, an explanation of why it exists and the derivation of its value. If it has been asserted by a sequence of rules, the variables and their values that were responsible for the rule firings are built into an english sentence describing their purpose. For example, the derivation of the variable representing the Remote Terminal Digital Loop Carrier might read "RT DLC Equipment Type is SLC Series 5 because CO Switch is 5ESS and State is New York." This audit trail, to which a user can append comments, serves to refresh the memory of the engineer at a later date, provide valuable insight to a novice engineer, or allow another engineer or a supervisor to follow the line of reasoning used to achieve a result. Variables can assume multiple instantiations in a given environment, which is useful when another dimension is needed to represent a slice of the information presented on a document. This also simplifies the resulting documents.

Menus never descend more than one level, reducing their complexity and the time it takes to instruct an engineer in their use. No longer will an engineer be confronted with an obsolete or inappropriate choice. As one of its more subtly intelligent techniques, OPERA provides dynamic menus which contain only relevant choices for a given situation. This means that the values provided in a menu for an item comprise the complete set eliminating the need for a user to, perhaps incorrectly, determine which choices are relevant. The same concept holds throughout OPERA, creating menus of items that are both necessary and sufficient.

No navigation through a myriad of screens is necessary in OPERA. One screen, containing a small number of windows, is displayed at all times. Included among these window are omnipresent command icons for all of the main functions of the system, and a one line window at the bottom of the screen that always contains a description of the functions of the three mouse buttons. The overriding premise is that, although a very detailed and well written user manual was available before the system was in beta testing, it would not be necessary. A user should be able to proceed, with a minimum of instruction, to engineer a system without any outside help.

Inferencing

OPERA waits for a new value to be entered and then it initiates its forward-based reasoning process. As values are entered on the screen, the inference engine evaluates them and determines if any of the nearly 2000 rules in the system are ready to fire and then fires them in priority order. Rules can have four types of actions: asserting values in fields, displaying relevant messages on a side-bar, displaying alert messages on an overlay window that suspends current operation and demands immediate attention, or calling up a different document.

Following is an example of an OPS-83 rule that selects the appropriate DLC equipment based on the current knowledge of the job:

```
rule DLC_SLC5_3
  -- Place a SLC Series 5 for some switches for NY
  {
   (Control Type=CLASS; Tag=PLAN;
            Status=ACTIVE);
   (Control  Type=RULE; Tag=DLC_SLC5_3;
            Status=ACTIVE);
   (Central_Office (@.Switch=|5ESS| || @.Switch=RSM ||
                    @.Switch=|AXE-10|));
   &1 (Equipment_Selection
        RT_DLC_Equipment <> |SLC Series 5|;
        RT_DLC_Equipment <> |LITESPAN-2000|;
        CO_DLC_Equipment_Origin <> AUDIT_USER;
        RT_DLC_Equipment_Origin<>AUDIT_USER);
   (Central_Office_Location State=|New York|);
   -->
     modify &1 (RT_DLC_Equipment = |SLC Series 5|);
     call imodify(index(&1), RT_DLC_Equipment,
                  |CO Switch,State|);
  };
```

The left hand side of each rule begins with two working memory elements to enable activation and deactivation of the rule at runtime by merely changing a text file. Other working memory elements provide the remaining conditions relevant for each rule. The right hand side will typically modify a working memory element and then call an OPS-83 function that notifies the system executive to modify the user interface's value for the associated variable. In this manner, rules assert values that are immediately redrawn on the document in front of the user. This particular rule will place a SLC Series 5 DLC device at the remote terminal when the central office has a 5ESS, RSM or AXE-10 switch and is located in New York. The user interface is told to modify its representation of the RT DLC Equipment Type and is told that the value was derived by a rule that fired because of the current values of the central office switch and the state. If, subsequently, the engineer changes the state in which the system is being planned or the switch located in the central office, or a rule asserts that the equipment will be a LITESPAN-2000, or the user overrides the system's recommendation, this rule will no longer match and attempt to assert a value.

Engineers currently need to select from among a large number of forms depending on the exact configuration of the system being designed. Here is a rule that is invoked when the user wants to display a plug-in diagram for the equipment that is selected on the planning form. The system determines which of the many plug-in forms is appropriate for the given conditions, relieving the engineer from the tedious bookkeeping:

```
rule DLC_Plug_In_4
  -- Select SLC Series 5 Plug In form for FPC universal
    applications
  {
   (Control Type=CLASS; Tag=PLAN;
            Status=ACTIVE);
   (Control Type=RULE; Tag=DLC_PLUG_IN_4;
            Status=ACTIVE);
   (System_Document
     Current_Document=|DLC Plug-In|);
   (Equipment_Selection
     RT_DLC_Equipment=|SLC Series 5|;
     Application_Type = Universal;
     (@.Mode=C || @.Mode=|C2| || @.Mode=D));
   -->
    call
      document(|SLC Series 5 Universal FPC Plug-In|);
 };
```

Rules leverage an engineer's effort by carrying the information entered or its derivatives to other parts of the current document or to other documents. As soon as it can, OPERA will begin to make determinations regarding the configuration and placement of equipment. When an engineer calls up another document, say the plug-in diagram of a multiplexer or digital loop carrier, it will already be mostly, if not entirely, completed. OPERA works under the philosophy that the engineer knows more than it does, so an engineer can always override the system, but the system cannot override the engineer. This means that OPERA will be of service to the engineer in laying a preliminary groundwork, but will permit the engineer to compensate for special or new circumstances. Maintaining the engineer's vital role in the operation is necessary for OPERA to help perform these fundamentally difficult engineering tasks. This will keep OPERA current, and casts it in the role of an assistant, not a replacement, for the engineer.

Rule maintenance is made easier by the existence of a text rule activation file that is read during system initialization. Although the rules are linked with the executable program, individual rules, or entire classes of rules can be activated or deactivated for a specific site, allowing us some latitude in the configuration of the system at a given site. Some fixes can be accomplished by deactivating rules, rather than forcing the users to await a future release of the software.

Database

A relational database resides transparently behind OPERA. A user can direct the system to save, load, delete or copy a job, which includes all variables with values and their derivations. In addition, the database contains information about sites and equipment, such as location codes, product codes and central office switches. Access functions permit OPERA to communicate easily with the database, regardless of its logical structure and contents. In the future, it is expected that OPERA will communicate with other systems through the sharing of data, which will remain transparent to the OPERA functionality. Control data in the database also exist to specify menu choices for variables and plug-in components for electronic equipment. Maintaining and selecting the information is thus much easier and more efficient than if it were residing in code or text files.

AI vs. Conventional Technologies

Traditional software systems were not able to address a significant portion of the outside plant engineering needs. The dynamic nature of the domain means that the system must be flexible enough to support frequent and rapid modifications. Using a knowledge based approach, where the program kernel is relatively small compared to the size of the knowledge base that drives the application, means that there is less overhead involved in keeping the system current, and the knowledge can be addressed in a more straightforward fashion. In addition, OPERA, which communicates with the user through documents that are similar in form and function to the current paper forms used in engineering, is intuitive and often times, self-explanatory, facilitating the knowledge encoding and validation processes. Providing similar functionality in a conventional software program would prove cumbersome, unwieldy and probably, untenable. The ability of our inference engine to explain itself, providing an important audit trail of decisions made by extracting information from each rule fired is an important tool for training engineers, refreshing an engineer's memory or verifying an engineer's work. Allowing random order entry of information on the documents permits the engineer to approach a problem in a manner relevant to the problem, not one determined by system design. Following multiple reasoning paths and weighting them appropriately permits an acceptable answer to be found in real time. Achieving these results with conventional techniques has not been possible in the past. OPERA utilizes these artificial intelligence techniques to successfully complete its tasks.

AI Techniques

OPERA exhibits intelligent behavior in the engineering arena as a result of long knowledge engineering sessions with very talented engineers. Knowledge is encapsulated in rules and in the sequences in which they are fired. However, knowledge is also represented on other levels. Help screens are a vital form of engineering knowledge, much as help from other engineers is a regular part of an engineer's day. Documents contain intelligence through conditions that dictate their form and content. They are designed schematically to graphically convey as much about the current task as possible. Multiplexers and digital loop carriers are represented with accurate pictures of their design, allowing service technicians to see exactly what the engineer intended. The intelligent user interface understands dates in any format, is case independent and re-formats letters and calculates appropriate line-breaks. These features combine to make OPERA a powerful tool that integrates with a knowledge base that is rich in engineering knowledge but that is independent and distinct in its design. OPERA is an intelligent forms-based expert system shell that is easily adaptable to any knowledge base to perform in a domain that has operations that are conducted through intensive handling of documents.

OPS-83

OPS-83 was selected as the inference engine to use because of its efficiency of both size and speed. It also is easily accessed from C, and thus became easily linked to our database and user interface components. Our reasoning paradigm did not require many levels of inferencing, nor were viewpoints or compound data structures necessary. Our needs were for a system that could efficiently handle a wide breadth of knowledge, accessed in small slices. Our decision tree is shallow but very wide. The remainder of our functionality is accomplished through C functions, which are easily called from within OPS-83 rules and which can easily modify OPS-83 working memory elements and initiate its rule firing.

System Integration

Seamless integration exists between OPERA and its underlying database, written in Oracle. Jobs can be saved and loaded by selecting a menu item in the command window. Each job is assigned a job key to make it unique. Job names are unique within a given location, corresponding to a central office serving area. Each variable that has a value is then saved in the relational database keyed by the job. Multi-value variables are further keyed by the index of the multi-value. Seamless integration also exists between the user interface component and the inference engine. As the user enters information on to a document, the corresponding working memory element in the production system, if one exists, is modified as well. The information that ties these two together is contained in the variable's data structure. Similarly, when a rule modifies a working memory element that corresponds to a variable, the variable value is modified and redisplayed if it is currently on the screen. These permit both the inference engine and the user interface to remain current in a manner that is completely transparent to the user. The connection between the two is specified in the configuration file that is read in at the start of an OPERA session and is easily modified without touching the program code.

A Novel Integration Of A Database, Knowledge Base and GUI

OPERA successfully integrates relational database technology with the latest in graphical user interfaces. It then ties these two technologies directly into a rule-based expert system component that provides the intelligence behind the actions of the system. Its design is not only effective, but efficient to implement for the developers. By displaying documents to our experts, we could come to an easier understanding of the task at hand. This method of successive refinement of the knowledge base while quickly displaying the results to the experts bridged the knowledge gap between the developers, who are computer scientists, and the engineers, who are telecommunications experts. The bottleneck that usually exists in such a scenario was alleviated by the schematic knowledge built into the documents and the easy to comprehend syntax that determines the properties of the variables and the functioning of the rules. We are even able to conduct knowledge acquisition sessions over the telephone, and a modem link. As the expert, sitting in front of an OPERA screen, suggests changes, we are able to modify the forms and variables that appear on the screen at the remote location without even ending the current OPERA session. The quick turnaround permits experimentation and demonstration, and removes the need to solve the entire problem all at once. Experts themselves, who each expressed a better understanding of their own domain of expertise after OPERA's introduction, became accustomed to trying out solutions to see how they worked, without fear that it would cause unnecessary and laborious effort to undo if it were not correct. Usually, their solutions were correct, and this flexible method was all that was required to help them cull the knowledge and put it in a workable form. The result is an expert system with an efficient methodology of development that underlies it, permitting quick integration of new knowledge and accurate and precise creation of a solution to, what had been, an elusive problem.

Application Use and Pay-Off

Application Use

The OPERA expert system has been deployed since July 1992 in engineering centers of New York and New England Telephone companies. Our system is used by the outside plant engineers of NYNEX, who plan and implement the installation of T1 Carrier equipment as well as Digital Loop Carrier devices. When a new job comes in, an engineer uses OPERA to do the planning and selection of equipment, and the necessary paperwork. Currently, the system is operational in 34 engineering centers throughout the New York and New England areas, assisting over 350 outside plant engineers. To the best of our knowledge, NYNEX is the first Regional Bell Operating Company to employ artificial intelligence techniques as a solution to both simplifying the outside plant engineering process and enhancing the skills of outside plant engineers.

Application Pay-Off

After the first three months of deployment, we conducted a survey to identify the benefits of the OPERA system. The benefits are categorized as performance enhancements, error prevention, equipment selection and plug-in inventory, and help provisioning. The results show a significant productivity enhancement. On the average, an outside plant engineer was able to finish his or her job at least three times faster using OPERA than using the manual engineering method. With more complex jobs, the time saving is even more pronounced - five to six times faster.

In the traditional manual way of engineering, documents were pre-printed forms where the engineers filled in the blanks for the necessary information. It was not unusual for these forms to be returned from the various processing offices due to incorrect, illegible or missing information, or even poor planning. As a result, a job took longer to complete and the redundancy contributed to a less competitive company. The forms produced using OPERA, on the other hand, are laser printed forms with noticeably higher quality and more consistent with the company's guidelines than the manual forms. Since the deployment, there have been no returned forms from any processing centers.

Each digital loop carrier device has from 60 to over 500 plug-in units that provide the required transmission and signaling functions. Depending on the type of terminal device and the current demand of service together with the forecast of future demand, an engineer has to determine the types of plug-in units and their correct number, and then populate the plug-in layout with the appropriate units. This task is extremely error-prone, time-consuming and tedious in the outside plant engineering process. It is therefore natural to have OPERA recommend the suitable terminal device and then calculate the optimal number of plug-ins, thus reducing the chances of installing the wrong equipment and eliminating the unnecessary inventory of plug-ins. In addition, the system selects the correct plug-in types and provisions them automatically. In the survey, this feature was rated as the most valuable feature.

The novice engineers discovered that by using OPERA they were able to finish their jobs without any assistance from the experts. In addition, the quality of their jobs was considered as high as those done by the experienced engineers. With the on-line help and audit trail facilities, OPERA is used as a tool to enhance the engineering skills.

Even though some benefits are not quantifiable due to insufficient data and lack of information, after careful financial analysis and review of the OPERA benefits we estimate that the system's contribution to the company as an outside plant engineering tool is up to five million dollars per year; 50% of this saving is contributed by performance enhancements, 20% by equipment selection, 10% by error prevention, 10% by plug-in inventory reduction, and 10% by providing help.

Application Development and Deployment

The development of OPERA started in July 1990 with one developer who did most of the ground-breaking work on identifying the domain and sketching the overall architecture of the system. Three months later, another developer joined the team. A prototype was then quickly assembled so that domain experts could easily feel at ease with the system while participating in the knowledge acquisition. The most challenging task of the development, surprisingly, turned out to be the identification of domain experts. Due to the diversity of the geographic make-up of the NYNEX regions (e.g., city areas like New York City, less urban areas like Newport, Rhode Island and rural areas like Portland, Maine), outside plant engineering methodology is followed differently depending on the make-up of the region. As such, the knowledge base of OPERA has to be composed of the skills of experienced engineers from the diverse geographic areas. After an extensive period of interviewing potential experts, we identified five top engineers throughout the NYNEX regions. The knowledge acquisition was an iterative process which continued throughout the development process. With the third member joining toward the end of the development phase, the team finished coding OPERA in 1.5 years, an equivalent of 3.5 man-years. The total development cost is estimated to be under $1 million.

Knowledge Acquisition

The OPERA team did not follow any one structured knowledge acquisition technique. Instead, we iteratively identified a piece (or some pieces) of knowledge; incorporated it with the rest of the knowledge base; and then refined, and validated the knowledge base. To do this, we followed these steps:

- Observation: We visited each expert and spent a couple of days at his or her office to observe and learn different functions that the expert has to perform daily. We even followed our experts to the field when they had to do site surveys. The purposes of this observation task are (1) to identify and organize the major processes of the outside plant engineering work model and their input and resources, and (2) to understand the foundations of the work flow in order to ensure that the end product, OPERA, will fit nicely in the operational environments.
- Interview: The observation gave us the necessary basic understanding of outside plant engineering so that we could pose intelligent questions to our experts. From our experience, the knowledge acquisition process becomes much more efficient when the developers know how to steer the experts in the right direction so that they could present their expertise in an resourceful manner. Similarly if the experts understand what the developers are seeking, the experts can help speed up the knowledge acquisition process.
- Validation: This step is needed to ensure that the

translation of the experts' knowledge was done correctly. The prototype was an extremely useful tool for the validation. After incorporating a new piece of knowledge into the knowledge base, we ran the prototype and asked the experts whether the system performed as expected. The experts were able to see the changes and validate the results.

System Validation

In January 1992, OPERA was ready for system test. The unit test and the integration test were completed in two months. The validation test, however, posed a more challenging task - to ensure the integrity of the information flow and to validate every rule in the knowledge base. A system test plan was designed to verify that OPERA's basic functions performed as expected and OPERA's rules fired appropriately. The domain experts then carried out the test cases, collected test results and compared them with the expected results. The system test lasted four months. After six months of total testing, OPERA was ready to be deployed.

Deployment

OPERA was deployed in stages. There are two reasons for this policy. The first is the limited resources. The second reason is that by having partial deployment at the beginning, we were able to put all of our efforts into discovering and correcting mistakes, studying the impact of OPERA's presence in the work flow and finding ways to ease the integration of OPERA into the operating environments. All of these issues are critical to the success of the full deployment. The partial deployment began in July 1992 and was completed in three months. The full deployment commenced soon after that and was finished in six months. Currently OPERA is operational in 34 engineering centers of the NYNEX telephone companies. All systems are running on the development platform which is a Sun SparcStation II. The deployment cost is estimated to be less than $20K per site which includes capital costs, technical support, and system training.

Training

New England Telephone and New York Telephone have different approaches to training engineers to use OPERA. In New England one of our top expert engineers, who has been working with us on the development and validation of OPERA, had become a full-time instructor at the Technical Training Center of New England Telephone. He then took the initiative to develop the training material for OPERA and incorporate it into the Digital Loop Carrier Engineering course. As a result, engineers who took this course were able to use OPERA even before the system deployment. In addition, at the beginning of the deployment, engineers were introduced to OPERA and had a chance to become familiar with the system in a three day OPERA training session.

In New York, the OPERA training course is an independent two-day course. An expert, who has been involved with the system development, wrote the training material and implemented the training procedure. The first half day of training is composed of the mechanics and functionality of the system, the second half is devoted to how best to use the system and the last day is spent on exercises and real-life examples. Engineers become proficient in using OPERA to do their jobs after three days of training.

Maintenance

Experience And Plans

The maintenance phase of a project is a set of software engineering activities that occur after a product has been turned over to the user. Planning for OPERA maintenance was performed concurrently with the normal project planning cycle. Maintenance is often required because of the need for enhancements due to new technology or modifications to the product to correct error conditions. Errors will occur and changes will be made in every system, regardless of how well it has been designed and implemented. OPERA maintenance focuses on the entire product and not on the source code alone. Documentation, for instance, will be a problem when changes are not reflected in design documentation, test documentation, or user manuals. Response of the support organization to user requests for changes or information should be quick and concise. The maintenance strategy for OPERA was put in place with all of the above considerations in mind.

The OPERA development team was fortunate in having excellent documentation, in the form of a comprehensive user manual, early in the life cycle of the project. The user guide was written by an experienced technical writer who took the time to examine the operating environment, talk to the users directly, and spend considerable time discussing the OPERA system with the expert engineers who helped construct our knowledge base. These same engineers that helped develop the system were also made available to train the people who are using it. A user who needs help can call a special telephone number and receive immediate assistance through the use of voice mail or electronic beeper. Our experience to this point has shown that users appear to be reluctant to read the manual or use the beeper, but prefer instead to speak to the developers personally. Plans that are being formulated now call for a first level of support that will not be a developer or expert engineer but a person who will filter the calls.

Simple problems, such as power cords or cables that have come loose, will be handled directly by the front line person. More serious problems, such as system crashes, printer problems, and bug fixes will be routed to a developer. Since each of the remote OPERA workstations has a 9600-baud modem and dedicated telephone line, a developer can dial in to any workstation quite easily to examine the logfiles, change the system configuration, or do whatever else is needed to diagnose and correct a problem. In fact, on several occasions, changes were made to the remote OPERA system while the user was still talking to the developer on the telephone! Updated versions of the complete OPERA system can be downloaded to a remote site in two to three hours. Minor changes can be made immediately. With a current deployment of 34 systems in widely scattered locations throughout six states modem access to each site makes support and maintenance manageable.

All changes to be made to the OPERA system require that a change request form be filled out and filed in the change management system. This form documents the nature and scope of the requested change, the responsible person or persons and, product version information. The OPERA change management form contains all of the fields needed for tracking purposes.

Maintenance Of The Knowledge Base

Understanding the differences from one DLC system to the next generation DLC system and integrating the new forms and rules into the existing OPERA system is a difficult and demanding task. The introduction of new rules to the knowledge base must always be accompanied by a review of the current rules to make certain that no conflicts exist. This is the reason that control of the knowledge base of OPERA will remain in the hands of the development group at NYNEX Science and Technology. As experienced expert system developers, we cannot take the risk of handing off the support of the OPERA system to anyone who doesn't have the expertise and understanding that went into building OPERA. Contact between the developers, the users, and the expert engineers will continue to assure the integrity of the OPERA system. Thorough testing of the system by the developers and expert engineers and iterative code refinement techniques have been employed in order to provide a more robust end product.

Quarterly meetings will be arranged with the user community to discuss changes and/or enhancements to the OPERA system. Any enhancements under serious consideration for implementation by the OPERA support group will be reviewed by subject matter experts before development begins. At this time we expect to provide one major and one minor software release per year.

Evolution Of Knowledge And Changes Over Time

Introductions of new digital loop carrier equipment and innovative techniques for managing the transmission of voice and data over telephone lines are constantly appearing in the Outside Plant engineering world. The OPERA system was constructed with the idea that it should be straightforward enough to integrate these innovations into the system without disturbing any of the existing logic. The LITESPAN-2000 system, for example, was introduced while OPERA was in the final phases of development and testing. LITESPAN-2000 is a new generation digital loop carrier system that provides a full range of POTS (Plain Old Telephone Service) and Special Services in a CSA (Carrier Serving Area.) A LITESPAN expert was identified and brought on board to provide LITESPAN expertise to the OPERA development team. Within a two month period the additional rules and forms for LITESPAN were created, integrated into OPERA, and tested in time for the scheduled OPERA deployment.

At this time DISC*S Sonet is in the process of being approved for use in Outside Plant engineering offices at New York Telephone. DISC*S Sonet is another next generation digital loop carrier system that can be fully integrated into existing telephone facilities. DISC*S Sonet will provide more capacity at higher speeds, and with greater reliability, than currently installed systems. A major step in digital loop transmission systems, DISC*S Sonet has its own microprocessor controlled supervisory system that provides the ability to perform system provisioning, channel provisioning, maintenance, inventory, performance monitoring, and diagnostics from the CO (Central Office) or the RT (Remote Terminal). An expert engineer has already been identified and work has begun on adding the capability to OPERA of designing digital loop carrier systems using DISC*S Sonet.

How The Design Facilitates Updates

OPERA is made up of many discrete pieces that work together to produce the result seen on the screen. When a screen displayable form is requested, it is created on the fly from an ASCII text file. New forms are simple to construct using our home-grown forms language and are refined by displaying them in the OPERA forms window as they are being built. Lines, boxes, rules, text, and variables with instantiating values are combined in an add-display cycle of successive refinement until the form achieves the desired result. In the same manner as the form layouts, variables from the configuration file, rules controlling the system responses, and schematic diagrams of various pieces of equipment can be customized to site requirements by editing local text files to enable or disable offending rules, forms, etc., and then recompiling the main OPERA executable module. The short compile time cycle of the OPERA executable means changes can be made and results can be evaluated quickly. The time element is especially important since we often make changes and test them remotely.

Conclusion

OPERA successfully leverages the expertise of the top engineers in New York Telephone and New England Telephone throughout the various engineering centers and dramatically improves outside plant engineering operations. It has met with wide acceptance by both the engineers who use it on a daily basis and high level management. OPERA also paves the way for advances in outside plant engineering systems.

At the Expert System Laboratory, we have gained valuable insight through our experiences with OPERA into the development and deployment of expert systems and into the operations of outside plant engineering in the telephone companies.

Acknowledgments

We would like to extend our appreciation to the subject matter experts that made OPERA possible: Ernie Horan, New England Telephone, Cathy Putzier, New England Telephone, Paul Buehler, New York Telephone and Steve Schwinn, New York Telephone. We would also like to acknowledge the efforts of our technical writer, Ivy Eisenberg and the director of the Expert Systems Laboratory at NYNEX Science & Technology, Jim Euchner.