AI Magazine Volume 13 Number 4 (1992) (© AAAI)

AI Research and Applications in Digital's Service Organization

Anil Rewari, editor, with Mark Adler, Peter Anick, Meyer Billmers, Mike Carifio, Alan Gunderson, Neil Pundit, and Mark W. Swartwout

■ The Digital Services Research Group and its predecessor groups and offshoots in Digital Equipment Corporation have been mobilizing leading-edge AI research to bear on real-life problems that face the corporation and its customers. The general strategy of the group is to explore emerging techniques relevant to service and support needs through developing rapid prototypes, deploying these prototypes, and incorporating feedback from users. With over 32 major projects undertaken during the past decade, we have worked on a broad spectrum of problems and explored a variety of advanced AI techniques. This article describes the current AI activities in five areas: (1) enterprise advisory systems, (2) natural language processing and textual information retrieval, (3) large-scale knowledge base management and access, (4) software configuration management, and (5) intrusion detection. We also list some future research directions.

For over a decade, Digital Equipment Corporation has been working at the leading edge of AI technology. Early work with OPS5 and the XCON system established Digital's reputation for building large, successful, deployed AI systems, and a plethora of additional work has been done in almost all areas of the field. Digital was active in the effort to specify Common Lisp (Steele 1989) and was involved in the first commercial implementation of Common Lisp; it is also currently involved in a research effort centering on the object-oriented language called TRELLIS. Knowledge representation work has led to participation in a consortium and the development of a frame-based representation system called IMKA TECHNOLOGY. The AI Technology Center (AITC) in Marlboro, Massachusetts, has been active in deploying knowledge-based solutions and training to corporate customers worldwide. AITC has been collaborating with the Microelectronics Computer Corporation (MCC) and leading universities in furthering research and its gainful deployment in industrial settings.

In 1981, a second AI group was founded at Digital Equipment Corporation that had a multidisciplinary focus under the technical direction of Neil Pundit.[1] The goal of this group, currently named the Digital Services Research Group, has been to mobilize leading-edge AI research to bear on real-life problems that face the corporation and its customers. Current research efforts focus on services, with the main group colocated at AITC in Marlboro and a smaller group in Colorado Springs, Colorado.

In 1982, the group used rule-based inference engines to build expert systems in specific domains. Even though early work used existing techniques, it also pushed the boundaries of technology. Our first expert system, called AI-SPEAR, was written in OPS5 and diagnosed faulty TU78 tape drives and controllers. Because the version of OPS5 at the time was not expressive enough for the problem, a new language, ALPHA, was invented. ALPHA was tailored to the problem, embedded in Lisp, and compiled into OPS5 (Billmers and Swartwout 1984). Using Alpha provided an 11:1 reduction in code mass and knowledge-acquisition time. This system initiated work on symptom-directed diagnosis at Digital's service organization.

OPS5 and the XCON system established Digital's reputation for building large, successful, deployed AI systems

We did not limit ourselves to expert systems. An early system called TEACH VMS provided help-desk functions to DECSYSTEM-20[2] users migrating from TOPS20 to VMS. Also, a workstation-based translator's assistant, TWS, exploited natural language processing to aid professional translators dealing with complex computer manuals in multiple languages (Carifio, Staub, and Anick 1988). Another system, SITA, addressed the problems in very large-scale integrated circuit layout in manufacturing by detecting and correcting errors in electronic elements that are laid on printed circuit boards (Simoudis 1989, 1990). The research for this project led to three patents.

PREDICTE, deployed commercially for an Australian construction firm, estimates time and cost for new construction. Rather than using an inflexible rule scheme, we deployed what was perhaps the first commercial use of constraint propagation, basing our engine on Guy Steele's seminal work (Steele 1980; Medoff, Register, and Swartwout 1988). Another system, CANASTA, was designed for the complex problem of analyzing operating system crashes. However, it also addressed more basic issues in building hybrid systems and the pragmatic issues of knowledge collection, validation, and distribution over several dozen support centers dispersed worldwide. PREDICTE and CANASTA, which were the fruits of our initial research efforts, won Innovative Applications Awards from the American Association for Artificial Intelligence (AAAI) (Stevens et al. 1990; Register and Rewari 1991).

University-sponsored work has always been important. Early funding from the group helped promote model-based reasoning at the Hardware Troubleshooting Group in the AI Laboratory at the Massachusetts Institute of Technology. We worked closely with Randy Davis and his students. Technology transfer of their approach to model-based diagnosis of digital circuits led to a system called ECLIPSE (Subramanian et al. 1989), which enabled us to troubleshoot failures in real-world situations based on circuit schematics. In addition, we have funded and collaborated with Stanford University's Knowledge Systems Laboratory, the MCC AI Program, Carnegie Mellon University on SOAR and diagnosis, Rutgers University, University of Pennsylvania, University of Texas, University of Southern California Information Sciences Institute, Yale University, and Brandeis University, among others. These externally funded research efforts have helped incorporate technology into our endeavors.

In our projects, the general strategy has been to explore the use of AI techniques in meeting service and support needs by developing rapid prototypes, deploying these prototypes, and incorporating feedback from users. This process proceeds in several iterations, with refinements or enhancements made to the prototype. If the prototyping effort is successful, we then undertake technology transfer to a product group that eventually builds a product.

Given the advanced nature of the investigation in many of our projects, we have been quite successful in our projects. As a research and advanced development group, we measure success in several ways. Clearly, prototyping efforts that result in eventual product development or deployment are considered successful. However, projects are deemed successful even though the product might not have been deployed if substantial learning takes place that directly influences follow-on projects.

Using this criteria, we analyzed 32 projects that our group has worked on since 1982 and that required more than 2 person-years of effort each. Three of these projects are still current, and it is too early to tell whether they will succeed. Of the remaining 29 projects, 23 (79 percent) have been successful, using the previous criteria for success. Four projects were terminated in an incomplete status because critical staff left or because the business needs of the corporation changed. Two projects failed.

Currently, we are engaged in five main areas of research: (1) enterprise advisory systems, (2) natural language processing and textual information retrieval, (3) large-scale knowledge base management and access, (4) software configuration management, and (5) intrusion detection. Work on enterprise advisory systems extends model-based reasoning into a new area that we believe will have important applications in help-desk and enterprise integration and also includes work in distributed AI. Work in large-scale, knowledge-based management and natural language–based information retrieval continues to push the boundaries of what is possible, trying to adapt it to what is useful to our customers and our support centers. Applications in configuration management and security intrusion detection demonstrate how AI research can lead to important service opportunities.

The rest of this article describes the AI activities in the five areas of investigation in services research. The last section gives a short synopsis of some major service projects during the past decade.

Research for Enterprise Advisory Systems

As computer systems become more complex, users become lost without a clear understanding of the whole enterprise. Users are often confused and need advice about how to perform even simple tasks, such as file transfers between different systems, and help to diagnose problems. The Enterprise Advisory System Team is working on a research project called OMNI (Billmers 1991). OMNI research is exploring ways to provide advice to users through a variety of techniques, including the use of domain models to provide high-level advice and determine if the problem can be delegated to other, more specialized knowledge-based systems.

Distributed AI

To provide for the integration of these diverse expert systems, a model of the expert systems must be provided, so that the problem is delegated to the appropriate system. The project team has adopted a distributed model along the lines of the work on contract nets initially described by Davis and Smith (1983). OMNI's approach is to create a wrapper around existing systems, including a model of the system's capabilities as well as the prerequisite input required for the system to operate. Thus, it allows the integration of existing expert systems without major modifications to their code.

The initial research used distributed AI techniques to link a number of expert systems that could explore the nature of the difficulties and benefits of such an approach to providing support (Adler and Simoudis 1990). The goal was to design an architecture to integrate otherwise disjoint problem-solving components without requiring their alteration. This task was achieved by associating a broker with each existing problem-solving component (*specialist*). The *broker* provides additional control mechanisms and decision support to incorporate each specialist as a modular unit conforming to the bidding and award cycles of the protocol.

The test bed linked a number of advisory and diagnostic systems and was successful in providing advice when problem domains were easily isolated. It did not require the expertise of more than one system. The problems of knowledge representation and knowledge sharing among heterogeneous knowledge-based systems is critical in this domain (Neches et al. 1991).

The OMNI project has been developing techniques to enable more cooperative behavior through the exchange of diagnostic information between knowledge-based agents. However, the main thrust of recent work has been to develop better models (and modeling tools) as a basis for advisory systems for a collection of components across an enterprise.

Micromodeling

A major problem in enterprise integration is overcoming inherent complexity. In an enterprise composed of n functional components, there might be n^2 (directed) interactions; as n grows larger, these interactions become the dominant consideration. Most computer-driven approaches to assist enterprise integration have been procedural in nature. Modeling requires that the integrator specify, in advance, components of the enterprise as well as their expected paths of interaction. Like the $O(n^2)$ nature of the search space, the number of these interactions becomes explosive with the growth of n. By contrast, a declarative approach to specify model behavior reduces the specification complexity to $O(n)$.

This declarative approach to model-based enterprise integration is called *micromodeling*. This approach draws on the AI literature in model-based reasoning (MBR), which seeks to describe systems in terms of the structure of the system and the function of its components (Davis et al. 1982; Genesereth 1982) and to reason from the model. Traditionally, MBR has focused on diagnosis and has begun with electronic circuits because of the well-specified behaviors of the components, the availability of precise structural descriptions, and the bounded sets of failure modes. Integrating an enterprise has interesting properties because of the interactions between functional components. It is assumed that these interactions can be specified using an input-output description of components (to represent structure) and procedural attachment to capture behavior.

The micromodel engine is driven by a set of classes arranged in a multiple inheritance hierarchy. At the top of the hierarchy are classes representing processing functions and processed entities, or data. By convention, processors are denoted as squares and data as circles; this notation is convenient for graphic portrayal of the objects and the search space. The processor class is defined as a set of input data and a set of output processed data. The input data are the resources that the processor needs to perform its function, which produces a set of output data.

The inference engine that searches the space of micromodels for a solution path is called PATHFINDER. PATHFINDER takes a starting object

A major problem in enterprise integration is overcoming inherent complexity

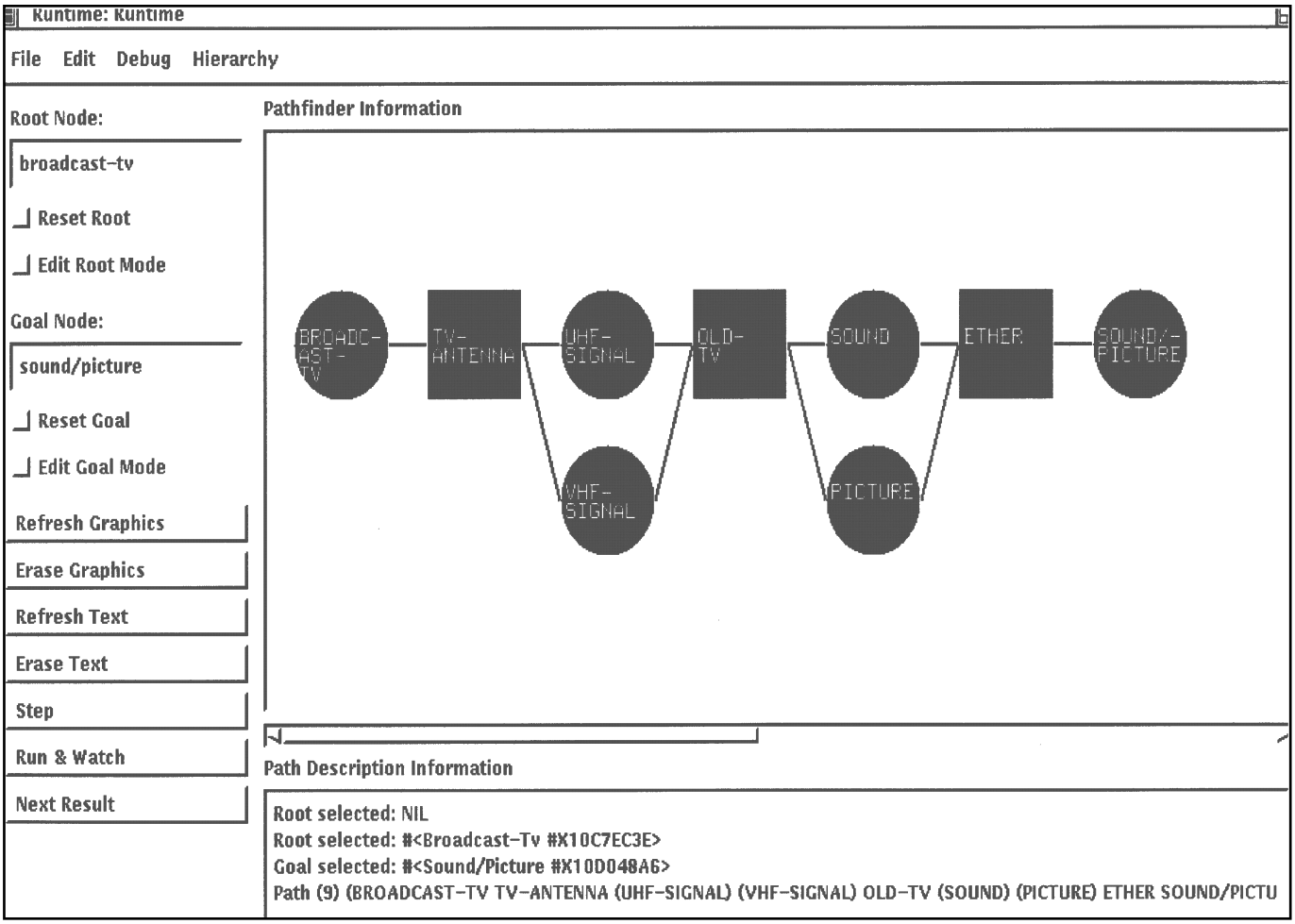

Figure 1. PATHFINDER *Run-Time Interface.*

Shown is an example of a multithreaded path caused by nodes OLD-TV and ETHER having Anded input in their micromodel descriptions. This causes PATHFINDER *to backtrack from a square. Such branching happens recursively and can be nested arbitrarily.*

The example shown here is a simple home electronics model, showing the paths through which signals flow. Circles represent signals, and squares are devices or components of a stereo system. By specifying starting and goal circles, the user can phrase queries such as, How do I record a broadcast FM program on my hi-fi VCR, or How do I connect my VCR to my stereo system so that I can listen to a video cassette in stereo? In general, PATHFINDER *can be used to find paths of interaction through any complex, enterprise-level system made up of components whose input-output behavior is simple. The solution path shows (possibly unanticipated) paths or the failure to find a path where one was expected. The textual path description information window can be used to show additional information about the path, such as constraints that were satisfied (or violated), total accumulated cost, or instructions on how to affect the square's behavior.*

and a goal object and tries to find a connecting path through the micromodel set. Constraints allow PATHFINDER to determine valid paths at run time. Constraint checking is done at run time, and context checking allows pruning of search based on inconsistent paths. An input constraint is code in the form of a predicate that is evaluated each time a circle is being provided by PATHFINDER as an input resource to a square. Input constraints allow the modeler to specify finer-grained, run-time checking of input in addition to class membership.

Function-preferences are a generalized constraint mechanism. A *function-preference* is a method defined on a processor class, a list of its input, and a list of its output. The ordered list of output returned is a (possibly proper) subset of the list of allowed output of the processor class. Like input constraints, function-preferences allow the modeler to write code to check slots and values of input. Solution paths can be nonlinear; that is, they can have branching. Figure 1 shows an example of a multithreaded path.

In the future, this team is interested in returning to the problem of integrating multiple knowledge-based systems to provide a wide range of advice. A number of technical problems to investigate have already been identified. Of principal interest is the problem of sharing partial solutions and concepts among heterogeneous agents representing different knowledge-based systems. One approach is to share a common base vocabulary

of high-level terms in a semantic network representation and to augment this base vocabulary with more specific domain knowledge for each agent. The shared high-level semantic representation allows for some exchange of information using the common vocabulary.

Participants: Mark Adler, Meyer Billmers, Meredith Malmberg, and Anil Rewari

Natural Language Processing & Textual Information Retrieval

Most of the knowledge available to the help-desk personnel at Digital's Customer Support Centers is currently in the form of natural language text—manuals, reports, memos, bulletin board postings, and so on. As over 20 years of research into the area of text storage and retrieval has made apparent, the task of finding relevant documents from a large set of online texts is complicated by the twin problems of mapping from a perceived information need to a computer-interpretable statement of the need and mapping the statement of this need to documents that might address it. Thus, many researchers view the problem of information retrieval as an iterative activity in which the users work with feedback from the system to progressively narrow the choice of documents that address their needs (for example, Belkin and Marchetti [1990]).

In the domain of computer troubleshooting, knowledge about technical terms and their correct use is often the key to the formulation of effective problem statements. Furthermore, users' confidence in search results depends on their understanding of how the system behaves in producing these results. A user must have a feeling for when it is worthwhile to continue pursuing an answer in a database and when it is likely that no relevant article exists online.

In response to such issues, the Intelligent Text-Retrieval Group has been investigating roles for natural language processing in full-text information retrieval. The group is developing the AI-STARS lexicon-assisted retrieval system (Anick 1992) in which canonical words and phrases are directly accessible through the user interface as units for manipulating query expressions. Specific subareas of research include morphological analysis, computer-aided lexical knowledge acquisition, and the integration of a natural language query facility with a direct-manipulation query-reformulation work space that incorporates explicit visual Boolean semantics. The group is also working in other research areas that although not directly characterizable as AI are nonetheless important components of the intelligent text-retrieval effort. These research areas include content-based data organization, data and query mapping across heterogeneous systems, object and index versioning, and data distribution (Anick, Flynn, and Hanssen 1991; Anick and Flynn 1992).

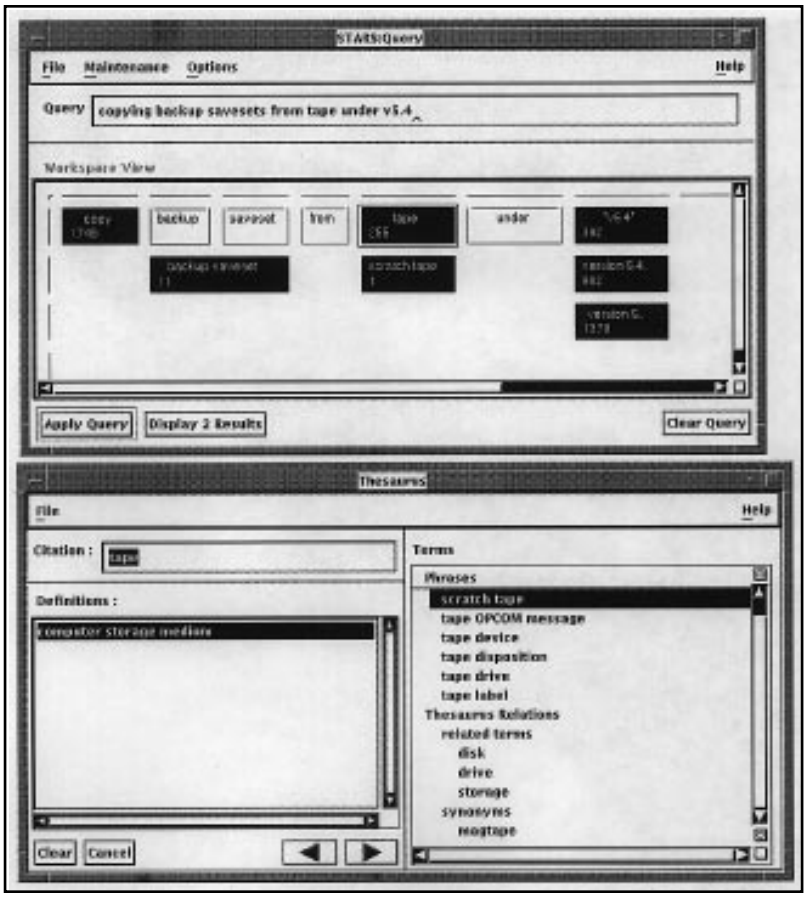

Figure 2. AI-STARS *User Interface.*

Query input window, and thesaurus window are shown.

Natural language query processing in the AI-STARS information-retrieval system involves mapping the words and phrases of a free-form query into a Boolean expression that can be evaluated by the retrieval system. The system exploits a direct-manipulation interface that reveals the results of any linguistic processing on the query and allows the end user the capability to reformulate the query with the (de)activation and movement of tiles representing the terms in the query (Anick et al. 1990; Anick 1991). As pictured in figure 2, the two-dimensional tile layout reflects a Boolean interpretation in which, roughly stated, terms along the horizontal dimension are Anded, and terms along the vertical dimension are Ored. New terms can be introduced to the query through both the natural language query window and a thesaurus window. (The underlying data structure for the query-reformulation work space is the same chart that is used by the system for bottom-up chart pars-

TEXMEX is a client-server application that provides uniform access to textual databases of various formats

ing of the input.) The current work reflects the group's interest in exploring practical near-term uses of natural language technology as well as laying the groundwork for the incorporation of more sophisticated linguistic inferencing. For example, were the system to attempt word-sense disambiguation or the generation of syntactic variants of larger phrasal units, it could also use a device such as the query-reformulation work space to make its inferences public and malleable.

To extend the capabilities of AI-STARS to other languages, the group has been developing a high-level language for the formal description of inflectional morphological paradigms (Anick and Artemieff 1992). The language attempts to capture, in a relatively straightforward way, the kinds of descriptions often presented in grammar books. It has separate rules to characterize orthographic and affixation behavior and makes use of a framelike inheritance mechanism to capture similarities and differences among paradigms. Descriptions expressed in this language can be compiled into data structures, which, when used in conjunction with a lexicon of canonical forms, support both the interpretation and generation of inflected forms.

The practical need to minimize the human effort involved in constructing knowledge bases for lexicon-assisted information retrieval has motivated the group's work on semiautomated lexical knowledge acquisition through the analysis of text corpora. The group has developed heuristics for mapping unknown word forms into the most likely inflectional paradigm and is utilizing statistics and phrase-structure rules to identify candidate multiword terms. In conjunction with a Brandeis University research effort to exploit generative lexicon principles (Pustejovsky 1991) for corpus analysis, the group is currently researching the use of *lexical conceptual paradigms*, systems of syntactic and morphological constructions that reflect underlying semantic categories, as a way of extracting semantic information about terminology for use in an online thesaurus (Anick and Pustejovsky 1990; Pustejovsky, Bergler, and Anick 1992).

Future work will continue the team's investigations into the use of natural language processing for multilingual information retrieval as well as explorations into the further incorporation of structured domain knowledge to facilitate the merger of textual information retrieval with expert system and case-based approaches to indexing and retrieval.

Participants: Peter Anick, Suzanne Artemieff, Rex Flynn, Jong Kim, Jim Moore, Mayank Prakash, Lalit Shahani, Clark Wright (Collaborators at the Colorado Springs Customer Support Center include Bryan Alvey, Norman Lastovica, Jeff Robbins, and James Wagner.)

Large-Scale Knowledge Base Management and Access

The Large-Scale Knowledge Base Management and Access Research Team is investigating the construction of persistent, large knowledge bases and their use by knowledge workers in various settings. This team has focused its attention on knowledge workers in the help-desk arena, but its work has wider applicability to the general office worker, that is, any environment in which knowledge is encoded in various ways (including less formal means such as semistructured text) and shared among knowledge workers.

This team is investigating some theoretical concerns, such as the expressiveness of various representations and the inference mechanisms that a representation can support. More importantly, it is investigating pragmatic concerns such as the unification of knowledge bases and databases, the population of knowledge bases with contents (ontology questions), and the refinement of these contents with use (knowledge acquisition, learning in several guises). The team hopes to augment existing databases with some knowledge to improve their utility. It is also investigating the distribution of knowledge through client-server–style applications.

Initial efforts have focused on the access and manipulation of text databases because text is what the average office worker today produces and consumes. Although at first glance this area might appear to have little to do with knowledge representation, this team is exploring methods and mechanisms for grafting access abstractions and domain knowledge, including a coarse-grained ontology, onto existing text databases in a heterogeneous, networked environment.

Over time, the team hopes to understand the trade-offs between highly structured, formal representations (such as semantic networks, which are amenable to machine processing but are a challenge to encode) and far less structured and formal analogs such as computer bulletin boards (where entries are easy to read and write but don't lend themselves to automated reasoning techniques). The group seeks to bridge the gap between the rich, powerful systems of tomorrow and the current corpus of corporate information that is encoded largely in text form.

The current step in this journey is a research prototype under development called

TEXMEX (text-management experiments). Several subsets of this problem have been attacked over the last two years, for example, in trying to classify and route customer mail messages (Register and Kannan 1992), providing uniform access to heterogeneous text databases, and augmenting the database contents with hypertext-style links.

TEXMEX is a client-server application that provides uniform access to textual databases of various formats, varied intentions, and varying capabilities. Figure 3 illustrates the architecture. In the diagram, three clients are connected to a knowledge server (called a resource server in the diagram). The knowledge server shields the clients from the physical details and location of the resources in the network (mail files, and so on) and provides a representational framework for all the resources this resource server knows; for example, a particular news group comp.lang.c++ (the mechanism) discusses the programming language C++ (the domain).

The current focus is on personal mail, news groups, and Digital VAXNOTES conferences that are used for personal correspondence and bulletin board–style discussion. The TEXMEX server provides locational transparency and implements the access abstractions. The server also encodes some domain knowledge about the topics covered in the textual databases and about interrelationships among the topics.

A range of clients are foreseen in this framework, from sophisticated tools for frequent users to focused tools for casual users. The project team has started with one particular client, named JEEVES. JEEVES is viewed as a kind of computational personal secretary that sorts, filters, and organizes incoming mail, news group, and VAXNOTES note traffic in the style popularized by Infolens (Lai, Malone, and Yu 1988).

The group hypothesizes that domain knowledge encoded in the server can be leveraged in several ways: Knowledge-based clients who manipulate incoming text with some semantic processing (such as classification), as well as syntactic processing, can be implemented. The team also hopes to investigate the discovery of new sources of text (new news groups, new VAXNOTES conferences) as these sources become visible in the network.

Participants: Meyer Billmers, Mike Carifio, Nari Kannan, and Michael Register

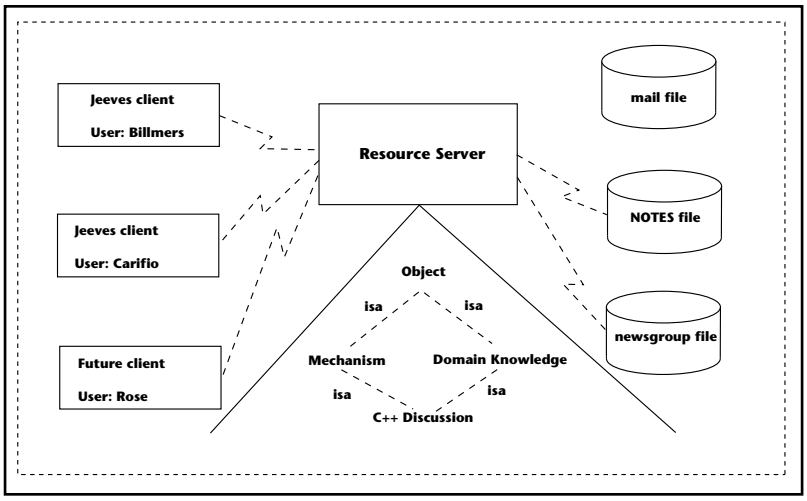

Figure 3. The Overall TEXMEX Architecture.

Shown are JEEVES clients accessing information sources through a resource server. The server contains resource-location information, resource-type information, and some domain knowledge about a resource's content (for example, some topical information).

Software Configuration Management Research

The Software Configuration Management Team has been investigating the use of knowledge-based approaches to address the software-revision compatibility and upgrade planning problem. In contrast with Digital's batch-oriented hardware compatibility system called XCON, the work on this project deals with the problem of software revision compatibility. The focus is on an interactive system that facilitates what-if analysis. Current systems have become increasingly complicated, so that there is a complex set of hardware and software dependencies that have to be met whenever upgrading a particular version of software or installing new software. Incompatibilities can result in system crashes or nonusable systems. The task of determining the impact of a proposed system revision is complicated and time consuming.

This team has developed a research prototype, CONMAN, to solve the revision compatibility and upgrade planning problem (Horner and Gunderson 1992). The primary knowledge representation is a semantic network of *n*-ary relationships residing atop an object-oriented inheritance hierarchy. An inference engine performs constraint satisfaction on this semantic network. An object-oriented blackboard (Adler et al. 1990) controls the incremental planning process.

Most of the research has concentrated on three key areas: knowledge representation, constraint-based incremental planning, and generalization representation clusters (GRCs). GRC is composed of a product description language (PDL), a PDL interpreter, and hash table data structures. The main purpose of GRC is the conversion of abstract domain knowledge into CONMAN's semantic network knowledge representation. GRC allows the manipulation of the semantic network at a higher level of abstraction rather than deal

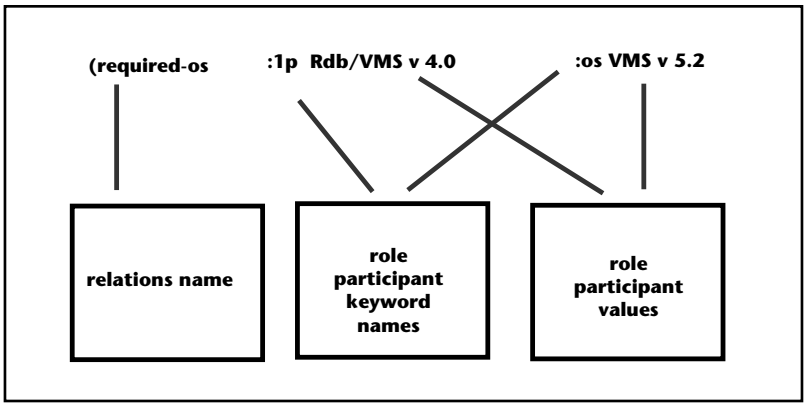

Figure 4. Product Compatibility Information for CONMAN *Is Specified with a Declarative Product Description Language.*

This language consists of a set of relational predicates, with the parts as shown in the figure. The example shows the specification for the operating system version that should be in existence on the computer system to use version 4.0 of the RDB/VMS *database management system.*

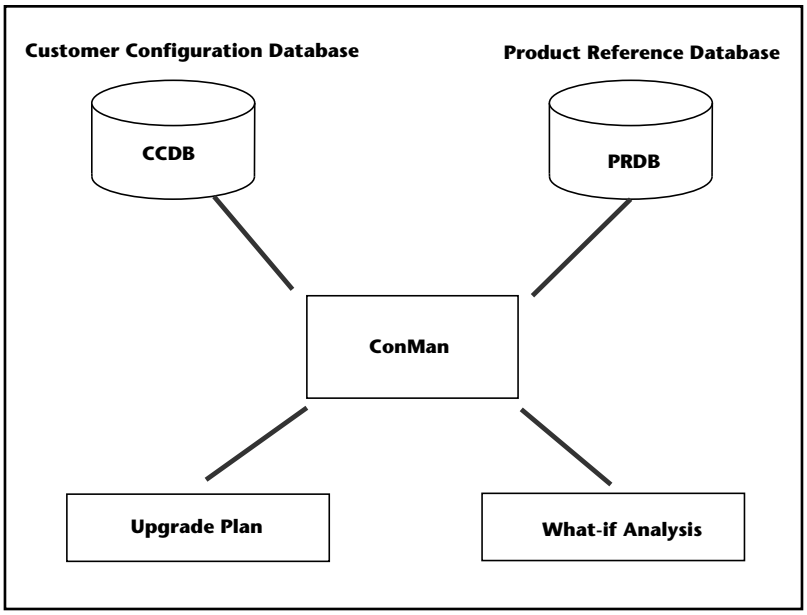

Figure 5. Working in a What-If Sandbox, CONMAN *Can Generate a Software Upgrade Plan for the User.*

The plan details the steps the user should follow to upgrade the software so that the entire computing environment is compatible and fully operational. For CONMAN *to generate this plan, a census of the software and hardware currently residing on a system is retrieved from the customer configuration database (*CCDB*). Software compatibility and constraint information from the product reference database is also extensively used.*

with knowledge representation implementation details. GRC is a generalization of earlier work called representation clusters–metaobject protocol (RC-MP) (Gunderson, Adler, and Schwartz 1990).

PDL allows declarative descriptions of product dependencies. Service engineers can also use this language to specify empirically derived product dependencies. For example, in figure 4, a required operating system predicate is depicted. Required-Os is the name of this CONMAN predicate. :LP and :OS are key words that define the role participant name. RDB/VMS V4.0 and VMS V5.2 are role participant values. This predicate translates into the following product dependency. The layered product RDB/VMS V4.0 requires the VMS V5.2 operating system.

Figure 5 depicts the architecture. CONMAN integrates knowledge and information from several sources. The main knowledge source is a relational database that contains generic software description information. Software descriptions include the prerequisite and dependency knowledge necessary for each product version. In addition, the appropriate product environment is also described. This database is the primary information source for CONMAN's product reference database (PRDB). The second source of information is customer system configuration information. This information consists of snapshot data of the hardware and software that are installed on customers' systems. With these information sources and external user information regarding layered products to be installed or upgraded, CONMAN makes appropriate inferences and produce a user-friendly upgrade plan. CONMAN also supports what-if analysis.

The research on CONMAN was just completed, and the current focus is technology transfer.

Participants: Alan Gunderson, Rosemary Horner, and Mark Adler

Knowledge-Based, Real-Time Intrusion Detection

Owners and users of computers must be concerned about security breaches by intruders and even about inadvertent damage by authorized users. To help assuage these fears, operating systems typically generate audit trails of events on the system, recording accesses to system objects and services.

Monitoring to protect the systems requires analysis of individual events in real time and the initiation of appropriate countermeasures. However, analyzing such an audit trail is tedious and can be difficult given a large volume of data (typically, several megabytes for each day for each system). When the analysis of records from many systems in a network environment is required, manual processing becomes impossible, and only spot checking is done.

One approach to intrusion detection relies largely on the statistical analysis of user behavior to identify uncharacteristic activity. A user's profile incorporates behavioral information such as the typical log-in time, the

average length of a session, and the average central processing unit time for each session. The user's activity is continuously compared to the profile, and the profile is updated. Activity that significantly deviates from the profile is flagged as anomalous (Lunt et al. 1992). This method requires a training period to establish a baseline and works best in a relatively static environment. A user can avoid detection by gradually modifying his or her behavior over time. Explanations for an alarm are not readily generated once an alarm is raised.

In contrast, the heuristic approach employed in this research uses general knowledge about security to analyze the audit trail produced by the operating system. Suspicious behavior is detected by recognizing actions that correspond to methods used by intruders. Suspect activity is tracked to collect evidence that an intrusion is real. Countermeasures are taken to prevent or minimize damage.

Using a knowledge-based approach achieves several tangible benefits. Because suspicion of an intrusion is based on the semantics of events seen, an explanation can be given when an alarm is raised. Also, innocent erroneous behavior is more easily recognized and discounted. The level of auditing necessary is minimized but can be increased dynamically when a situation warrants it.

The Intrusion Detection Team has built a real-time security monitoring prototype for the ULTRIX operating system, called ESSENSE. The ESSENSE prototype monitors the audit-record stream generated at the system service level and recognizes higher-level, security-relevant actions. To recognize the user actions that generated a stream of audit records, these records must be viewed in the context in which they occur. Related actions are identified and grouped into sets, representing a stream of logically connected events. A rule base analyzes the sets of events and generates responses. The prototype uses an object-oriented approach to the representation of events, objects, and cases. This approach allows the use of a generic description of the items of interest, with specializations as needed.

The ESSENSE prototype detects actions that might be attempts to subvert the security policy of an installation and collects auxiliary information necessary for making decisions. It communicates significant activity to system management and can take countermeasures directly.

An important future generalization of the ESSENSE approach is to look at the correlation of events across hosts to detect an attack on the network, as shown in figure 6. This feature requires representing the network as a system of connected and interrelated hosts. Certain types of events might be relevant to the network as a whole, but others might not. Another promising extension is to use temporal reasoning techniques to improve the discrimination based on the temporal ordering of events. Furthermore, it seems that the approach taken in the ESSENSE prototype is applicable to other areas, such as fraud detection in a transaction-processing environment. There, as in computer security, a trail of records can be analyzed for their security relevance.

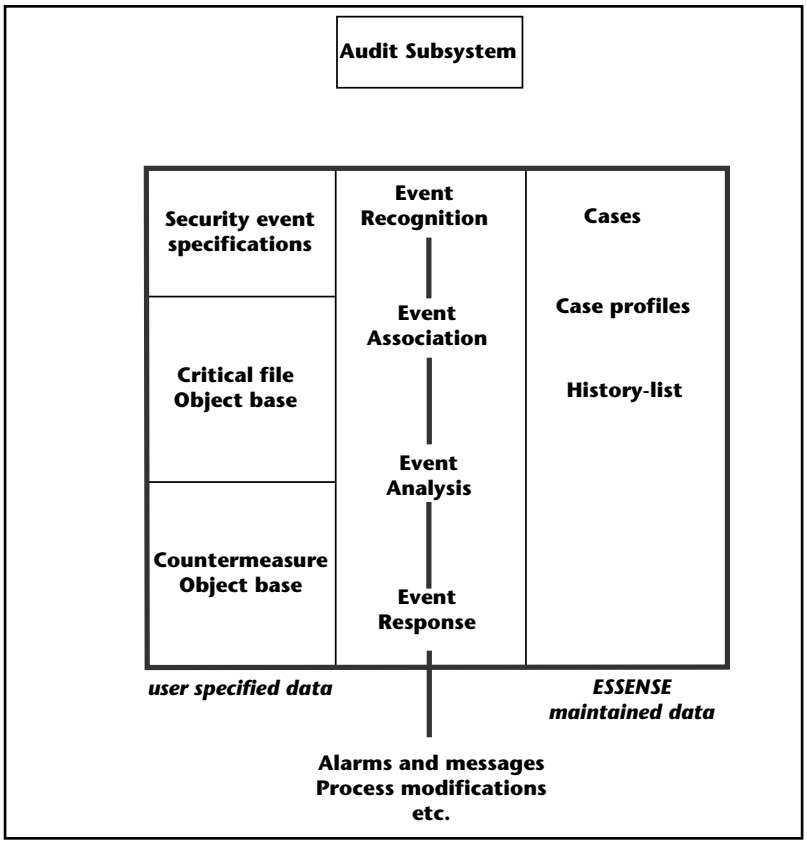

Figure 6. ESSENCE *Uses General Knowledge about Security to Analyze the Audit Trail Produced by the Operating System.*

Suspicious events are detected and countermeasures taken to prevent or minimize damage.

Participants: Gary Hoglund, Ed Valcarce, and Mark Swartwout

Future Research Directions

The research group in Digital Services continues to expand its work in new directions, applying new technologies to productive use within services. Certain AI-related areas have great potential. Research in machine learning

can lead to the automatic generation of new knowledge or the adaptation of existing knowledge to reduce the brittleness of knowledge-based systems in the service organization. Topics that can be explored include generating rules from case databases, using inductive machine learning to help index cases stored in case databases, and using inductive machine learning to form categorizations of various articles. Case-based reasoning is another area of promising research. Much of the problem solving at the service centers draws on past experience with similar cases. Work would focus on the more difficult problem of adapting retrieved cases, that is, the modification of a standard case to more appropriately fit the problem at hand, as well as indexing cases. Research in real-time AI is also important to Digital Services in areas such as network monitoring and diagnosis, detection of security violations, and predictive diagnosis of software and hardware.

Research efforts also need to focus on certain challenging problems, solutions to which can save our service organization effort and costs. Research in designing systems for serviceability could eventually lead to big savings in servicing and provide more reliable systems. As hardware becomes a commodity item, the major focus is on software reliability. Research on facilitating code reuse and providing facilities to select and assemble modules with known behaviors to build complex applications can result in more reliable applications. However, given that software malfunctions are unavoidable, research in software diagnosis and ways to circumvent software misbehaviors is important. Research in knowledge sharing can lead to service organizations tapping into engineering knowledge bases. Finally, AI techniques can potentially play an important role in harnessing the use of emerging computing technologies, such as pen-based computing, voice-based computing, and wireless communication-based computing.

Major Services Research Projects over the Past Decade

Digital Services Research has worked on over 40 AI projects since its inception in 1982. Most of these projects were targeted toward Digital's own service and support centers as opposed to external customers. Besides these projects, our group has also been involved in providing consulting services and training to external customers. Below is a list of some of the major projects that our group has worked on during the past decade. Many of these projects subsume work done on earlier projects, and several projects are not listed because of business reasons.

AI-SPEAR: This project, undertaken from 1982 to 1983, diagnosed faulty TU78 tape drives and controllers using rules. This system initiated work on symptom-directed diagnosis at Digital's service organization and is still widely used for diagnosing certain types of hardware failures.

AI-STARS: This project involves applying advanced technologies, including natural language processing, object-oriented database techniques, and direct-manipulation user interfaces to the storage, retrieval, and administration of Customer Service's technical information databases. This effort, initiated in 1987, has four patents pending. Work on this project is still proceeding and is described in this article.

CANASTA (crash analysis troubleshooting assistant): This system assists personnel in the analysis of operating system crashes, one of the most complex problems seen at the support centers. CANASTA provides several types of problem-solving modules; incorporates sophisticated remote automatic data collection; clusters unresolved cases into similarity sets to assist experts in discovering new rules; and provides for the collection, validation, and distribution of rules from a network of support centers worldwide. Research started in 1988, and the system was deployed even as a prototype worldwide in Digital's support centers. It has been made into a product and is in routine use. It was awarded an Innovative Applications Award by AAAI in 1991.

CONMAN (configuration manager): This decision support system provides an action plan of steps that a user can follow to install or upgrade software-layered products on his or her system. Started in 1989, the project led to a prototype that is currently being made into a product. This project is described in more detail in this article.

DECTREE: DECTREE is a tool that allows nonprogrammers to create, modify, build, and execute decision tree–based applications through a graphic user interface. The project was started in 1989. It is currently being used to build applications at many of our service and support centers and is available to customers under consulting arrangements.

ESSENSE: This system monitors computer security alarms in real time and takes countermeasures against suspected intrusions or malicious acts. The project was started in 1990, and the system was successfully demonstrated and evaluated as a prototype.

FOXGLOVE: This rule-based language, layered

above Lisp, was developed in our group. It was used successfully by us and others in Digital from 1985 to 1991 for rapid prototyping of applications. It is no longer used.

LANALYST (LAN analyst): This project investigated the use of AI for network diagnosis. It used a structural model of the components and network connectivity of a local area network. Reasoning was driven by activity models that described network function, for example, message sending (comprising the subactivities transmit, relay, and receive message). Although LANALYST was not made into a product, much was learned from this project. It has one patent pending.

OMNI: This project uses model-based reasoning to address customer queries for advice or diagnostic assistance online. Through a technique called micromodeling, it provides a means for describing how the components of an enterprise fit together, providing a model that can be used to give advice. It is part of our current research efforts and is described in this article. This project has one patent pending.

PREDICTE: One of our nonservice-related projects, it estimates the time of constructing a multistory building. It uses constraint-propagation techniques. It is widely used by the Australian customer for whom it was developed. This project was awarded an Innovative Applications Award by AAAI in 1990.

SITA (signal integrity and troubleshooting assistant): This project addressed issues of detecting and correcting errors in electronic elements that are laid on printed circuit boards. Work on this project led to three patents; one is currently pending. A prototype was delivered.

SKIS (skill identification system): This text-classification application takes natural language text as input and assigns a classification to the text from a predefined set of classifications. A prototype application was developed that parses incoming customer service requests and classifies the text into 1 of 130 different technical skills. The classification is then used to route the service request to the appropriate technical support specialist within the Customer Service Center. This project has one patent pending, and we are exploring technology transfer to a product group.

TEACH VMS: This system was an early help-desk application for migrating TOPS20 users to VMS. Developed in 1985, it used a rule base.

TELESWAT: This groupware tool consists of an enhanced *X* server that allows a user in one location and one or more users in other locations to view and access shared windows and operate cooperatively in the shared environment. The system facilitates group working sessions at our support centers. It is deployed at a few sites. Work is proceeding on this project.

TWS (translators' workstation): A prototype was developed between 1986 and 1988 that provided a customizable workstation for Digital translators in Europe trying to translate Digital English-language documents into other languages. It provided online help for dictionary and morphological analysis. Although this prototype was not made into a product, much of the research effort transferred into the AI-STARS project.

ULTRADOC: This expert system for UNIX system administration manages a set of workstations using various UNIX variants. The expertise consists of experiential knowledge, as well as straightforward knowledge, for strict compliance with detailed specifications. It is currently being developed as a prototype and will be deployed at a few test sites soon.

Acknowledgments

We want to acknowledge the encouragement and guidance of Digital Services through John Shebell, corporate consulting engineer, who has acted as our mentor since the early 1980s. Senior officers of the corporation have provided sponsorship and guidance throughout these past 10 years.

Notes

1. The first AI group was the XCON group formed in 1978.

2. DECSYSTEM-20, TOPS20, VMS, VAXNOTES, ULTRIX, and RDB/VMS are registered trademarks of Digital Equipment Corporation.

References

Adler, M. R., and Simoudis, E. 1990. Integrating Distributed Expertise. In Proceedings of the Tenth International Workshop on Distributed AI. Austin, Texas: MCC.

Adler, M.; Davis, A.; Weihmayer, R.; and Worrest, R. 1990. Conflicting Resolution Strategies for Nonhierarchical Distributed Agents. In *Distributed Artificial Intelligence*, volume 2, eds. L. Gasser and M. N. Huhns, 139–161. London: Pitman.

Anick, P. 1992. Lexicon-Assisted Information Retrieval for the Help Desk. Paper presented at the Workshop on Artificial Intelligence for Customer Service and Support, Eighth IEEE Conference on Artificial Intelligence Applications, Monterey, California, 3 March.

Anick, P. 1991. Integrating "Natural Language" and Boolean Query: An Application of Computational Linguistics to Full-Text Information Retrieval. Paper presented at the AAAI 1991 Workshop on Natural Language and Text Retrieval, Anaheim, California, 14–19 July.

Anick, P., and Artemieff, S. 1992. A High-Level Morphological Description Language Exploiting Inflec-

tional Paradigms. In Proceedings of Computational Linguistics 1992, volume 1, 67–73. Nantes, France: International Committee on Computational Linguistics.

Anick, P., and Flynn, R. 1992. Versioning a Full-Text Information-Retrieval System. In Proceedings of ACM/SIGIR '92. New York: Association of Computing Machinery. Forthcoming.

Anick, P., and Pustejovsky, J. 1990. An Application of Lexical Semantics to Knowledge Acquisition from Corpora. In Proceedings of Computational Linguistics 1990, volume 2, 7–12. Helsinki, Finland: International Committee on Computational Linguistics.

Anick, P.; Flynn, R.; and Hanssen D. 1991. Addressing the Requirements of a Dynamic Corporate Textual Information Base. In Proceedings of ACM/SIGIR '91, 163–172. New York: Association of Computing Machinery.

Anick, P.; Brennan, J.; Flynn, R.; Hanssen, D.; Alvey, B.; and Robbins, J. 1990. A Direct-Manipulation Interface for Boolean Information Retrieval via Natural Language Query. In Proceedings of ACM/SIGIR '90, 135–150. New York: Association of Computing Machinery.

Belkin, N. J., and Marchetti, P. G. 1990. Determining the Functionality and Features of an Intelligent Interface to an Information Retrieval System. In Proceedings of ACM/SIGIR '90, 151–177. New York: Association of Computing Machinery.

Billmers, M. 1991. OMNI: A Heterogeneous, Distributed Advice Giver. In Proceedings of the Seventh Conference on AI Applications, 160–163. Washington, D.C.: IEEE Computer Society.

Billmers, M., and Swartwout, M. W. 1984. AI-SPEAR, Computer System Failure-Analysis Tool. In Proceedings of the European Conference on AI, 65–73. Amsterdam: Elsevier.

Carifio, M.; Staub, L.; and Anick, P. 1988. Translator's Workstation Technical Description, Technical Report AIAG-TWS88-1, Digital Equipment Corporation, Marlboro, Massachusetts.

Davis, R., and Smith, R. G. 1983. Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence* 20:63–109.

Davis, R.; Shrobe, H.; Hamscher, W.; Wieckert, K.; Shirley, M.; and Polit, S. 1982. Diagnosis Based on Description of Structure and Function. In Proceedings of the Second National Conference on Artificial Intelligence, 137–142. Menlo Park, Calif.: American Association for Artificial Intelligence.

Genesereth, M. 1982. Diagnosis Using Hierarchical Design Methods. In Proceedings of the Second National Conference on Artificial Intelligence, 278–283. Menlo Park, Calif.: American Association for Artificial Intelligence.

Gunderson A.; Adler M.; and Schwartz S. 1990. Implementing Representation Clusters with a Metaobject Protocol for Model-Based Network Diagnosis. Paper presented at the Third CLOS Users and Implementers Workshop, Ottawa, Ontario, Canada, 21 October.

Horner, R., and Gunderson, A. 1992. CONMAN—A Software Upgrade Planner. Paper presented at the Workshop on Artificial Intelligence for Customer Service and Support, Eighth IEEE Conference on Artificial Intelligence Applications, Monterey, California, 3 March.

Lai, K.; Malone, T. W.; and Yu, K. 1988. OBJECT LENS: A "Spreadsheet" for Cooperative Work. ACM Transactions on Office Information Systems 6(4): 332–353.

Lunt, T. F.; Tamaru, A.; Gilham, F.; Jagannathan, R.; Jalali, C.; Neumann, P. G.; Javitz, H. S.; Valdes, A.; and Garvey, T. D. 1992. A Real-Time Intrusion-Detection Expert System, International Technical Report, SRI Project 6784, SRI International, Menlo Park, California.

Medoff, S.; Register, M.; and Swartwout, M. W. 1988. Representing Knowledge for Design Verification and Evaluation Systems. In *Artificial Intelligence Developments and Applications*, eds. John Gero and R. Stanton, 135–145, Amsterdam: Elsevier.

Neches, R.; Fikes, R.; Finin, T.; Gruber, T.; Patil, R.; Senator, T.; and Swartwout, W. R. 1991. Enabling Technology for Knowledge Sharing. *AI Magazine* 12(3): 36–56.

Pustejovsky, J. 1991. The Generative Lexicon. *Computational Linguistics* 17(4): 409–441.

Pustejovsky, J.; Bergler, S.; and Anick, P. 1992. Lexical Semantic Techniques for Corpus Analysis. *Computational Linguistics* (special issue on corpus analysis). Forthcoming.

Register, M., and Kannan, N. 1992. Combining Knowledge-Based and Statistical Techniques for Text Classification. In Proceedings of the Fourth International Conference on Tools for Artificial Intelligence. Washington, D.C.: IEEE Computer Society. Forthcoming.

Register, M., and Rewari, A. 1991. CANASTA: The Crash Analysis Troubleshooting Assistant. In *Innovative Applications of Artificial Intelligence* 3, eds. R. Smith and C. Scott, 195–212. Menlo Park, Calif.: AAAI Press.

Simoudis, E. 1990. Learning Redesign Knowledge. *IEEE Transactions on CAD*. 9(10): 203–226.

Simoudis, E. 1989. A Knowledge-Based System for the Evaluation and Redesign of Digital Circuit Networks. *IEEE Transactions on CAD*. 8(3): 302–315.

Steele, G. L. 1989. *Common Lisp: The Language*. Burlington, Mass.: Digital Press.

Steele, G. L. 1980. The Definition and Implementation of a Computer Programming Language Based on Constraints, Technical Report, MIT TR-595, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Stevens, G.; Stretton, A.; Register, M.; Medoff, S.; Fung, M.; and Swartwout, M. 1990. PREDICTE—An Intelligent System for Indicative Construction Time Estimation. In *Innovative Applications of Artificial Intelligence 2*, eds. A. Rappaport and R. Smith, 81–98. Menlo Park, Calif.: AAAI Press.

Subramanian, K.; Billmers, M.; Sitterly, S.; and Baker, P. 1989. Fault Diagnosis in Complex Digital Circuits. In Proceedings of the Fifth Conference on AI Applications, 159–164. Washington, D.C.: IEEE Computer Society.

Mark R. Adler is a software principal engineer in the Digital Services Research Group. He received B.S. degrees in mathematics and computer science from the Massachusetts Institute of Technology and his Ph.D. in AI from the University of Edinburgh, Scotland, in 1977. His research at Digital has focused on ways to improve enterprisewide services, including the use of distributed AI techniques and the use of symbolic models to represent the capabilities of components for advice and diagnosis. He chaired the distributed AI workshops at the Ninth and Tenth National Conferences on Artificial Intelligence.

Peter Anick is a software principal engineer in the Intelligent Information Applications Development Group, where he heads the AI-STARS and INFO BUILDER efforts. His work on industrial applications of computational linguistics has included database and expert system interfaces, machine-aided translation, and full-text information retrieval. He is currently pursuing a Ph.D. at Brandeis University, working on generative lexicon theory and its applications.

Meyer Billmers has a B.S. from Carnegie Mellon University and an M.S. from Harvard University. Following school, he became a research staff member at the AI Laboratory at the Massachusetts Institute of Technology (MIT). He joined the Digital Services Research Group at its inception 10 years ago and has been active in many of the research projects. He is also technical liaison to the MIT AI Laboratory. His research interests include model-based reasoning, intelligent online advisory systems, and automatic interpretation of semistructured information.

Michael Carifio is currently a software principal engineer in the Digital Services Research Group. He holds B.S. and M.S. degrees from Boston College and Boston University, respectively. He is the project leader of the TEXMEX project and directs the knowledge representation investigations.

Alan Gunderson is a software principal engineer in the Digital Services Research Group. Alan holds an M.S. in computer science from Northeastern University. His undergraduate degree in computer science comes from the University of Montana. He is the author or coauthor of numerous publications in the areas of expert system development, knowledge representation, and software engineering. He is a member of the Association for Computing Machinery, the Institute of Electrical and Electronics Engineers Computer Society, and the American Association for Artificial Intelligence.

Neil D. Pundit is a group engineering manager and heads the Digital Services Research Group. He holds a B.S. from the Bihar Institute of Technology, India, an M.E. in electrical engineering from Texas A&M University, and a Ph.D. in electrical engineering from Auburn University. Most of his personal technical contribution has been in the guidance and control of space vehicles, for which he was awarded the NASA/TRW Lunar Landing Award in 1969 and the U.S. Air Force Association's highest honor in science and engineering, the Theodore von Karman Award, in 1975 as part of the Viking Flight Team for the Mars landing missions at the Jet Propulsion Laboratory–California Institute of Technology. From 1977 to 1980, he was the director of technical activities at the headquarters of the Institute for Electrical and Electronics Engineers. In 1980, he joined Digital. He is the founding technical manager of the service's AI group and has been influential in the choice and direction of all the projects.

Anil Rewari is a principal software engineer in the Digital Services Research Group and holds an M.S. in computer science from the University of Massachusetts, Amherst, and a B.Tech. in electrical and computer engineering from the Indian Institute of Technology, New Delhi. He chaired the workshop on AI for customer service and support at the IEEE Conference on AI Applications in 1992. His interests include distributed AI, model-based reasoning, and machine learning.

Mark W. Swartwout, who holds a B.S. from the University of Illinois and an M.S. from Indiana University, has been with the Digital Services Research Group since the beginning. He was project leader or manager for the AI-SPEAR, CANASTA, and PREDICTE projects, among others.