AI Magazine Volume 12 Number 3 (1991) (© AAAI)

An Overview of Some Recent and Current Research in the AI Lab at Arizona State University

Nicholas V. Findler and Uttam Sengupta

■ *This article contains a brief description of recent and ongoing projects in the AI Lab at Arizona State University. Every project represents a team effort: either direct involvement of the participants or intellectual contribution from all other members of the AI Lab.*

The AI Lab was established within the Computer Science Department in 1982 when Nicholas Findler arrived at Arizona State University. The lab is now fairly well equipped, with a DEC 11/780, six VAXstation 3100s, a Texas Instruments Explorer II, an Apollo 4000 workstation, and several MAC IIs. Through a universitywide network, researchers also have access to a number of large-scale mainframe systems, including large DEC, IBM, and CRAY machines. In addition to our team, several other faculty members and their students in other labs are also engaged in various AI-related activities. Their work is not included in this description.

The Quasi-Optimizer System

The quasi-optimizer (QO) system automatically analyzes and synthesizes decision-making strategies. In other words, it is capable of generating a computer model (descriptive theory) of an observed strategy and, from several strategy models, a new strategy that is better than any other single one (a normative theory in the statistical sense). The observation can be done in either the passive mode, where the system does not or cannot interfere with the environment, or the active mode, where the system designs a sequence of environments for the decision maker to respond to.

Applications of the QO system include the automatic and guided acquisition of knowledge from human experts for expert systems, the automatic training and evaluation of control operators, and the automatic verification and validation of discrete-event simulation models. Figure 1 shows the relationship among the major modules of the system.

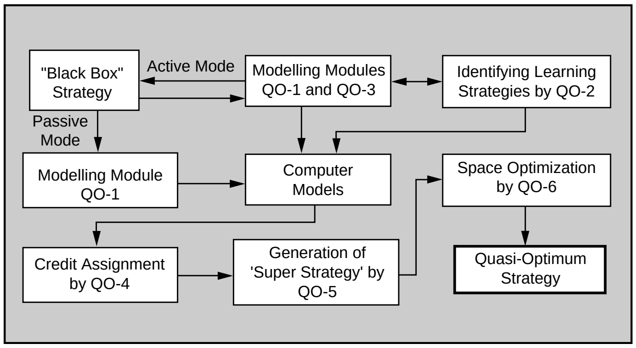

Figure 1. The Relationship between the Six Major Modules of the Quasi-Optimizer System.

Participants: Neal Mazur, Robert F. Cromp, Bede McCall, Mike Belofsky, Tim Bickmore, Jan van Leeuwen, João Martins, George Sicherman, and Nicholas V. Findler.

The Advice Taker-Inquirer

The advice taker-inquirer (AT/I) is a domain-independent program used to construct, fine tune, and oversee the operation of an expert system. It consists of two phases: (1) a learning phase, during which a human expert interactively teaches the system about a domain in terms of principles, high-level examples, rules, and facts, and (2) an operational phase, during which the program monitors the expert system as it is applied to its actual environment and continually attempts to improve its performance by hypothesizing new rules and reorganizing existing knowledge.

Figure 2 shows the principal components of this system. The applications include the user-advised construction of an assembly line balancing system and a self-optimizing street light control system.

Participants: Robert F. Cromp, George Sicherman, Steve Feuerstein, Skip Lewis, and Nicholas V. Findler.

The Generalized Production-Rule System

The generalized production-rule system (GPRS) can estimate or predict values of hidden variables. (*Hidden variables* can only be observed and measured intermittently, at irregular points of time and space, in contrast with *open variables,* whose values can be identified at any time and location.) The estimation is based on generalized production rules expressing stochastic and potentially causal relations between known values of hidden variables and certain mathematical properties of open-variable distribution. A multidimensional learning process adds to, consolidates, and optimizes the generalized rule base. The system can serve as a module of an expert system in need of numeric or functional estimates of hidden-variable values. An extended version can predict not only point values but also the functional forms of the hidden variables; it can also handle geographically distributed input operations and knowledge bases.

An application of the system has been in the area of forecasting and interpolating econometric indicators. Figure 3 illustrates the types of basic morphs used to fit open-variable distribution.

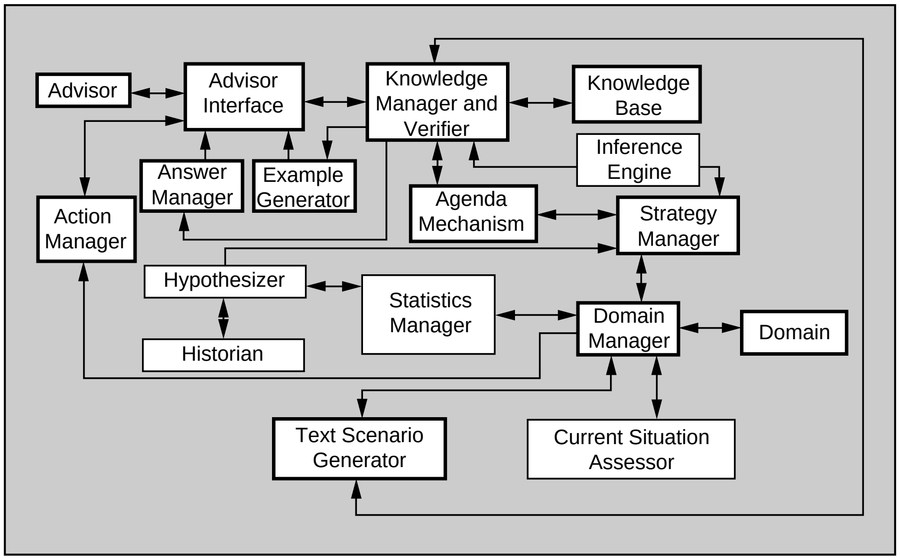

Figure 2. Block Diagram of the Principal Components of the Advice Taker-Inquirer System.

Those active during the learning phase are marked with heavier lines.

Participants: Ron Lo, Cher Lo, Ernesto Morgado, John E. Brown, Han Yong You, and Nicholas V. Findler.

Causal Modeling System

CMS and NEXUS aim at discovering causal relations between events—CMS in engineering-scientific domains and NEXUS in socioeconomic domains in which no well-defined domain theory is usually available.

CMS fuses into a single knowledge base (the *causal graph*) different pieces of information obtained from observing the environment (empirical derivation statistically corroborated), interacting with humans (guided learning from relational and narrative material), and using the laws of nature (analytic derivation). Such inexact, incomplete, probabilistic, and fuzzy information is the basis on which the system then generates hypotheses about causal relations. These hypotheses can be reinforced, modified, or rejected as new information becomes available. The system can answer questions concerning possible causal relations that also satisfy various temporal, spatial, and other restrictions. Figure 4 illustrates the knowledge sources for CMS.

NEXUS's input information is gleaned from texts in a pseudonatural language form, which it first converts to a conceptual dependency format. The reasoning process combines commonsense knowledge with the spatiotemporal conditions of causal relations. The learning component acquires new knowledge and changes the interpretation of prior knowledge without human guidance; episodes (events, plans, goals, objects, and relations) are merged in its memory as causal connections and similarities between them are identified. A causal graph is constructed and, with more information acquired, gradually enlarged. The system uses this structure to respond to user queries about causal relations between events.

Participants: Tim Bickmore, Cem Bozsahin, and Nicholas V. Findler.

A Predictive Man-Machine Environment

The predictive man-machine environment (PMME) is a useful tool for training and evaluating human decision makers and planners. It can also serve as the basis for routine operations. The user is seated in front of two graphic display units. One shows the characteristic features of the current world and is updated in a time-driven, user-driven, or event-driven manner. The other display unit shows the extrapolated world, indicating the expected consequences of the user's tentative decisions prevailing at a future time point. If the user is satisfied with the expected consequences, s/he finalizes his(her) decisions. Otherwise, s/he goes through the cycle of decision-extrapolation-evaluation as many times as necessary, time permitting. The quality of his(her) decisions are automatically evaluated, and information about the strengths and shortcomings of his(her) decision-making strategy is fed back to him(her). We have used this environment to train and evaluate novice air traffic controllers. Figure 5 shows the two displays of the PMME system.

Participants: Tim Bickmore, Bob Cromp, Neal Mazur, Min-hui Kuo, Congyun Luo, and Nicholas V. Findler.

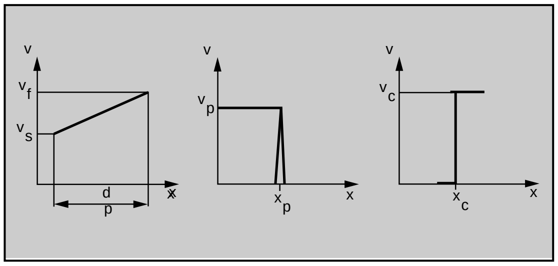

Figure 3. The Three Types of Basic Morphs Used to Fit Open-Variable Distribution—Trend, Sudden Change, and Step Function— Each with Its Characteristic Parameters.

There can be a period over which the mathematical description of the open-variable behavior is not possible with the required level of statistical significance, called "delay function."

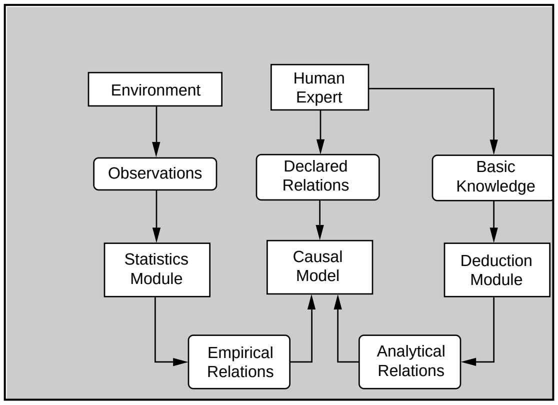

Figure 4. Knowledge Sources for the Causal Modeling System Used in Engineering-Scientific Domains.

Figure 5. The Two Displays of the Predictive Man-Machine Environment System Showing Simulated Radar Scopes.

The left-hand side displays the current world and the right-hand side the extrapolated world. The controller's microphone is used to provide voice input to the simulated pilots.

A System for Heuristic Retrieval of Information and Facts

A system for heuristic retrieval of information and facts (SHRIF) makes use of, rather than simulates, the human strength and flexibility in dealing with incomplete, irrelevant, redundant, and inconsistent information. Information is organized in a hierarchical fashion. Semantic networks connect atomic and composite nodes, each representing an item (object, concept, event, action, or category) and a multitude of its descriptors. Bidirectional edges connect the nodes and are labeled with syntactic and semantic restrictions and strengths of association (degree of similarity) to enhance the traversal of the network.

The user can select one of several available algorithms to form clusters of nodes that share certain implicit commonality. Further, planes contain certain clusters of some similarity. Retrieval requests, which might be fuzzy, incomplete, or even inconsistent with the contents of the knowledge base, are first compared with automatically generated cluster and plane labels. From this comparison, the system generates a list, ordered in decreasing plausibility, of clusters and planes that might contain the item most satisfying the search criteria. The area of application used in developing the system was the retrieval of medical information. Figure 6 illustrates the design schema of SHRIF.

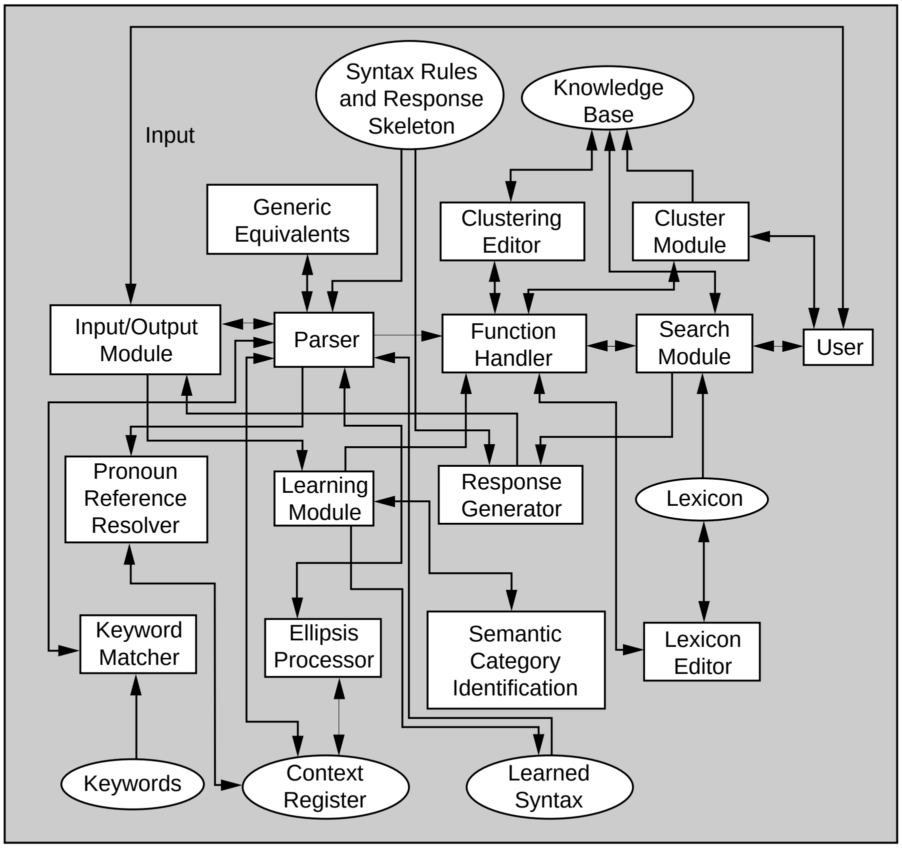

Figure 6. The Design Schema of the System for Heuristic Retrieval of Information and Facts.

Participants: Sarita Maini, Albert Yuen, and Nicholas V. Findler.

Computational Lexicometry

Computational lexicometry (CL) is a branch of dictionary science that deals with the mathematical and statistical aspects of dictionaries. The project is concerned with the ways of recording meaning and with the coverage by a basic set of unit meanings in monolingual dictionaries. Conjectures have been made about the mathematical relationships between the size of the covered set, the size of the covering set, and the maximum length of dictionary definitions. The results of a series of systematic experiments, using an existing dictionary of computer terminology, verified the conjectures. Besides the inherent interest in these investigations, the concepts established and the methods of quantifying dictionary variables might eventually lead to more efficient dictionaries with respect to precision, compactness, and the computing resources needed for processing. Figure 7 shows relationships between dictionary descriptors.

Participants: Heino Viil, S.-H. Lee, and Nicholas V. Findler.

Distributed Planning and Problem-Solving Systems

Distributed planning and problem-solving systems (DPPSS) handle tasks that cannot be dealt with effectively and efficiently by one single processor. Such tasks are characterized by geographically distributed input-output operations, a large degree of functional specialization, time-stressed demands for solution, and a need for reliable computation and graceful degradation in case of the partial failure of components. The theoretical issues studied in this project relate in general to problem decomposition, subproblem distribution, subproblem solution, and synthesis of the total solution. Five domains of applications are described in the following subsections as separate project developments.

Participants: Ron Lo, Qing Ge, Ji Gao, John Stapp, Curt Chapman, Uttam Sengupta, Cem H. Bozsahin, Serban Catrava, Raphael Malyankar, Parag Parekh, Francis Annareddy, and Nicholas V. Findler.

Studies on Distributed Air Traffic Control

A location-centered, cooperative planning system based on a network of loosely coupled nodes (airborne processors) was developed. Nodes belong to decentralized, dynamically formed (possibly overlapping) subgroups. The knowledge base is distributed, and the control structure among the subgroups is hierarchically organized and is based on nominator–coordinator–co-worker relations. At the same time, a node can assume several roles as a nominator, coordinator, or co-worker at different levels. Connections between coordinators and co-workers are made through mutual selection and with minimal communication. The planning paradigm adopted is simulation-based *incremental shallow planning*, a multipath and multistage search process. The information structure that supports planning, *distributed scratch pads*, enables a self-healing process in case of the partial failure of processors and communication channels. A series of experiments have been carried out in a general-purpose distributed AI test bed to measure the respective levels of control performance in three organizational structures: a local centralized architecture and a location-centered cooperative planning system with a one- and two-level coordinator–co-worker hierarchy. Figure 8 shows the kernel design of the individual airborne processors.

Participants: Ron Lo and Nicholas V. Findler.

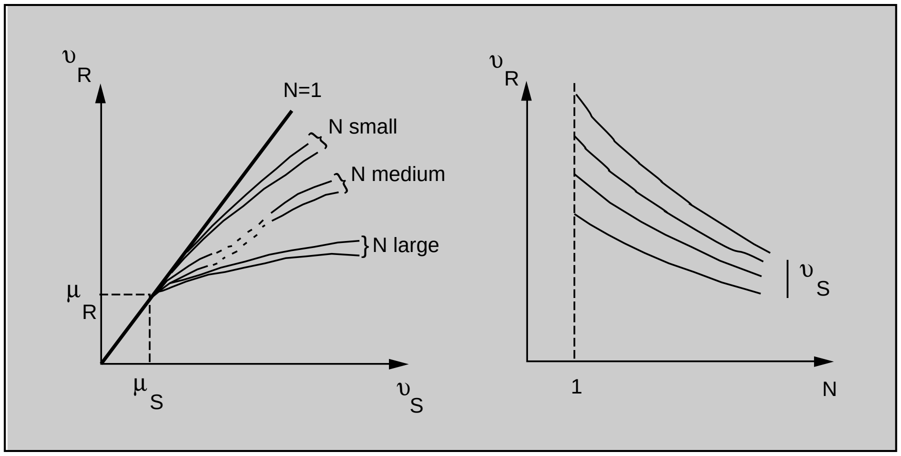

Figure 7. Asymptotic Relationships between Dictionary Descriptors: The Size of the Covering Set (νR) and the Covered Set (νS) and the Maximum Possible Length of the Definition String, N.

Distributed Hierarchical Control of Nationwide Manufacturing Operations

The general paradigm comprises a group of dissimilar plants distributed geographically. Each plant is connected to a transportation and a computer network of limited capacity and varying reliability. The constituent plants possess both spatial specialization (in terms of their geographic location and connectedness) and functional specialization (in terms of equipment, capabilities, skills, and cost functions). The environment is dynamic: New orders come in, and the operational availability and the cost functions of manufacturing, transportation, and computational resources change unexpectedly. There are two possible criteria of operation. The combine of all plants has to produce a given number of final products at either minimum overall cost within a given period of time or a given cost figure within a minimum period of time. Either of these criteria requires an optimum allocation schedule of the processing assembly operations to individual plants over space and time and the satisfaction of a set of given and estimated constraints. The flow of control and information between the decentralized, loosely coupled collection of problem solvers is defined by the manufacturing process of given products. The negotiation process enables a node to reveal as much of its status information to selected others as necessary. Network perception keeps a node's model of the network up to date based on messages sent and received. Figure 9 illustrates the design of the distributed control of nationwide manufacturing operations.

Participants: Qing Ge, Ji Gao, Glen Reece, and Nicholas V. Findler.

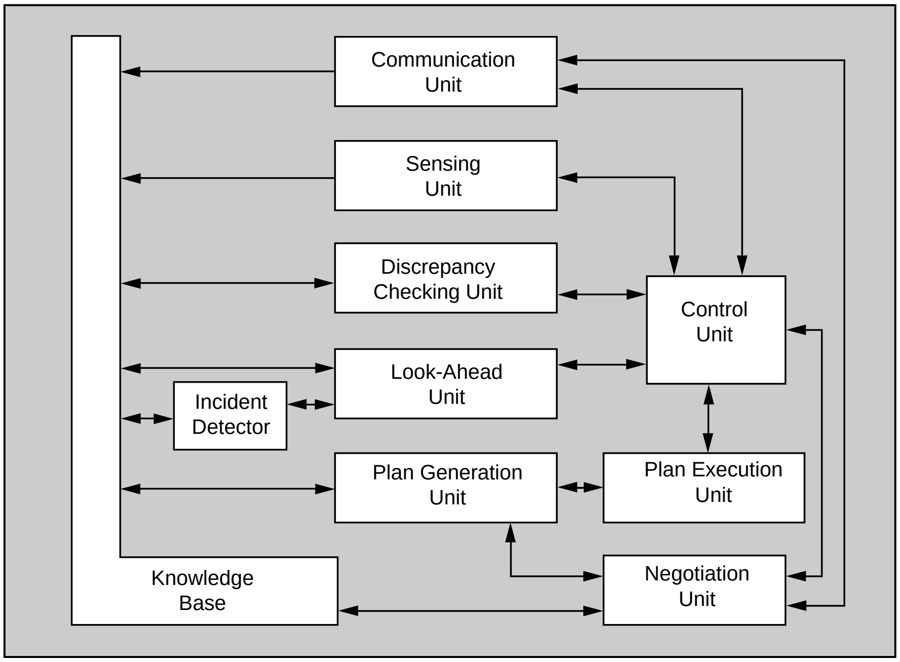

Figure 8. The Kernel Design of the Individual Airborne Processors.

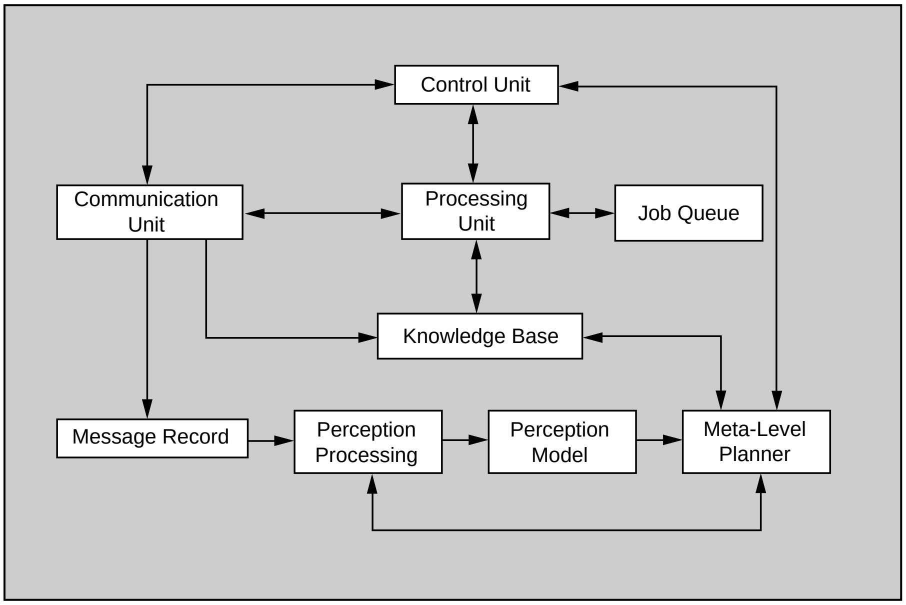

Figure 9. Design of the Distributed Control of Nationwide Manufacturing Operations.

Distributed Control of Street Traffic Signals

This project is about the control of traffic signals by a network of distributed processors situated at street intersections. Every processor runs an identical expert system (with a somewhat different knowledge base) and communicates directly with the four adjacent processors. Messages can also reach indefinitely distant processors, modulated by the needs of intervening ones. The information transmitted can be raw data, processed information, or expert advice. The rule base of the expert systems has a natural segmentation, corresponding to different prevailing traffic patterns and the respective control strategies. Multidimensional learning programs optimize both the hierarchy of the rules and the parameters embedded in individual rules. Different measures of effectiveness can be selected as the criterion for optimization. Traffic scenarios are automatically generated for a characteristic period: a certain part of the day (for example, morning rush hours), a certain type of day in the week (for example, regular work day), or a certain season of the year (for example, vacation time). The results of our first implementation—a running prototype simulation program—has proven the feasibility and utility of the approach. Figure 10 shows the Manhattan grid used in distributed street traffic control.

Participants: John Stapp, Curt Chapman, Serban Catrava, and Nicholas V. Findler.

Distributed Control of Time-Stressed Allocation of Moving Resources to Moving Tasks

There is a hierarchy of decision-making entities, each with an area of jurisdiction and a set of moving resources of different types. Scheduled and unscheduled tasks of different priorities and types need to be serviced by an appropriate resource mix. The problem is how to allocate the resources over space and time for optimum task accomplishment while a set of constraints is satisfied. Four fundamental issues are being studied within this paradigm:

First, we are considering the relative measures of effectiveness of strict hierarchical versus flexible latticelike structures for command, control, and communication activities.

Second, the constant ownership of resources by the entities is replaced by the idea of *dynamic scoping* when, as a result of lateral communications, resources are loaned or returned to or from the nearest qualified node within the current scope in response to declared needs.

Dynamic scoping refers to situations in which a particular planning agent does not have enough resources to respond to the tasks in its area of jurisdiction and invites adjacent agents to loan resources they can spare. In case these agents are not able to satisfy the request either, they contact further adjacent agents for help, and so on. In turn, when tasks are accomplished, the loaned resources are returned to their original owner.

Third, soft real-time planning is performed in nondeterministic environments when some plan components have strong precedence requirements and others do not, and some actions have to be completed before a given time point and others have more relaxed time constraints. Accordingly, a priority number is associated with each action being planned; this number is the product of constant factors (for example, the action's absolute and relative importance) and varying factors (for example, the action's current time criticality). The priority numbers control the ordering of the planning process objectives.

Fourth, a hierarchical planning mechanism creates skeleton plans that are then refined into instances of

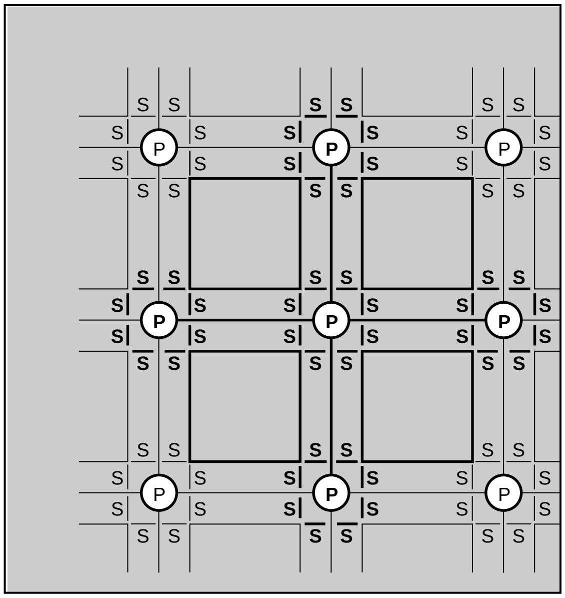

Figure 10. The Manhattan Grid Used in the Simplified Prototype Model of Distributed Street Traffic Control.

A processor, P, at every intersection receives input data from its own sensors, S, and various types of information from the processors at the adjacent four intersections.

action sequences and resource mixes by a rule-based system using a time-critical inference mechanism. The rules are arranged in a hierarchy of priorities so that more and more details are taken into account with each iteration of the inference mechanism. This ordering guarantees that the result of the planning process is always as good as possible within the time available. The planner is supplemented with a simulation model for plan verification and monitoring.

Participants: Uttam Sengupta, Cem H. Bozsahin, Raphael Malyankar, Parag Parekh, Francis Annareddy, and Nicholas V. Findler.

Intelligent Distributed Simulation Models

The fusion of simulation modeling and knowledge-based decision support techniques is important because the combined system can gain from the power and specialization of the components. Simulation is ideal for modeling and understanding complex phenomena, and knowledge-based systems provide the cognitive support—a property that simulation models lack. However, the marriage of these two diverse techniques introduces the problems of communication and the need for a unified representation scheme. Performance also suffers because both these techniques are computationally expensive. The plan generator for multiple unit law enforcement simulation is a knowledge-based planning system driven by a sophisticated simulation model. It is designed to measure the performance of United States Coast Guard resources under dynamically changing conditions. It serves as a tool for planning resource schedules and patrol routes and enhancing and measuring resource utilization. Results from these studies can also be used in phasing out less reliable and less useful resources and acquiring new ones. We are studying ways in which the total computational task can be divided into geographically or functionally distinct subtasks, which are then performed on several connected workstations. Figure 11 shows the user front end of the law enforcement simulation system.

Participants: Raphael Malyankar, Uttam Sengupta, Parag Parekh, Francis Annareddy, and Nicholas V. Findler.

Figure 11. The User Front End to the Law Enforcement Simulation System for the United States Coast Guard Project.

An AI Excursion into Social and Cultural Anthropology—An Automated Discovery System to Identify Rules for Inheritance, Succession, Marriage, Injunction against Incest, and Exogamy

This project deals with the problem of discovering rules that govern social interactions and relations in preliteral societies. Two computer programs were constructed some years ago that can receive data, possibly incomplete and redundant, representing kinship relations among named individuals. The programs then establish a knowledge base in the form of a directed graph that the user can query in a variety of ways. Another program, recently written on top of these older programs, can form concepts of various properties, including kinship relations, of and between the individuals. The concepts are derived from the examples and counterexamples of a certain social pattern, such as inheritance, succession, marriage,

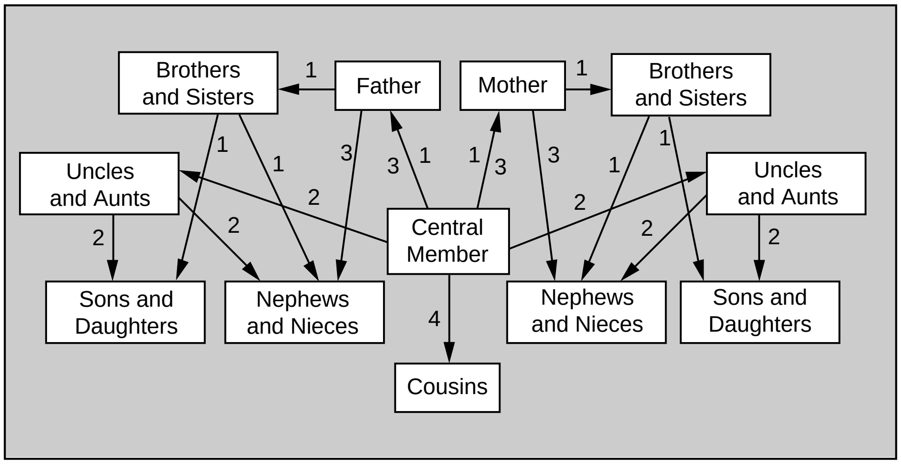

Figure 12. An Exemplary Retrieval Pattern: The Four Categories of Pathways from the Central Member to All Cousins.

class (tribe, moiety, clan, and so on) membership, domination-subordination, incest, and exogamy. The concepts become hypotheses about the rules, which are corroborated, modified, or rejected by further examples. Figure 12 shows an example of an exemplary retrieval pattern.

Participants: W. R. McKinzie and Nicholas V. Findler.

Acknowledgments

The projects were supported in part by the National Science Foundation, the Air Force Office of Scientific Research, the U.S. Department of Transportation, the U.S. Coast Guard, Digital Equipment Corporation external research grants, and the CACI Products Company.

Bibliography

A complete bibliography and copies of individual papers can be obtained directly from N. V. Findler, Department of Computer Science and Engineering, AI Lab, Arizona State University, Tempe, AZ 85287.

The Quasi-Optimizer System

Findler, N. V. 1990. Contributions to a Computer-Based Theory of Strategies—Chapter 3, The Quasi-Optimizer (QO) System, 13–41. New York: Springer-Verlag.

Findler, N. V.; Bickmore, T. W.; Ihrig, L. H.; and Tsang, W. W. 1990. A Note on the Comparison of Five Heuristic Optimization Techniques of a Certain Class of Decision Trees. *Information Sciences* 53:89–100.

Findler, N. V., and Mazur, N. 1989. On the Automatic Verification and Validation of Models. *The Transactions of the Society for Computer Simulation International* 6:153–172.

Findler, N. V.; Tsang, W. W.; and Ihrig, L. H. 1987. A Pruning Algorithm for Finding All Optimal Decision Graphs. In Proceedings of the International Conference on Optimization: Techniques, and Applications, 470–479. Singapore: ICOTA.

Findler, N. V., and Cromp, R. F. 1984. A Heuristic Approach to Optimum Experimental Design. *Computational Statistics and Data Analysis* 2:167–178.

Findler, N. V., and McCall, B. B. 1984. A Conceptual Framework and a Heuristic Program for the Credit-Assignment Problem. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-14:750–754.

Findler, N. V. 1983. An Overview of the Quasi-Optimizer System. *Large-Scale Systems, Theory and Applications* 5:123–130.

Findler, N. V. 1983. Automatic Analysis and Synthesis of Strategies: A New Branch of Artificial Intelligence. In Proceedings of the IEEE Phoenix Conference on Computers and Communication, 239–245. Washington, D.C.: IEEE Computer Society.

Findler, N. V. 1983. On Automatic Generation of Descriptive and Normative Theories. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 596–601. Washington, D.C.: IEEE Computer Society.

Findler, N. V.; Belofsky, M. S.; and Bickmore, T. W. 1983. On Some Issues Concerning Optimization and Decision Trees. In Proceedings of the International Conference on Mathematical Modelling in Science and Technology, 191–197. Oxford: Pergamon Press.

Findler, N. V.; Mazur, N.; and McCall, B. 1983. A Note on Computing the Asymptotic Form of a Limited Sequence of Decision Trees. *Information Sciences* 28:207–231.

Findler, N. V. 1982. A Preliminary Report on a Self-Optimizing Experimental Design-Generator. In Proceedings of COMSTAT82—Fifth International Symposium on Computational Statistics, 70–77. Würzburg, Federal Republic of Germany: Physica Verlag.

Findler, N. V., and Martins, J. P. 1981. On Automating Computer Model Construction—The Second Step toward a Quasi-Optimizer System. *Journal of Information and Optimization Sciences* 2:119–136.

Findler, N., and van Leeuwen, J. 1979. The Complexity of Decision Trees, the Quasi-Optimizer, and the Power of Heuristic Rules. *Information and Control* 40:1–19.

The Advice Taker-Inquirer

Findler, N. V. 1990. Contributions to a Computer-Based Theory of Strategies—Chapter 4, The Advice Taker/Inquirer (AT/I), 43–90. New York: Springer-Verlag.

Findler, N. V.; Sicherman, G.; and Feuerstein, S. 1979. Teaching Strategies to an Advice Taker/Inquirer System. In Proceedings of the EuroIFIP-79 Conference, 457–465. Amsterdam: North-Holland.

The Generalized Production-Rule System

Findler, N. V. 1990. Contributions to a Computer-Based Theory of Strategies—Chapter 5, The Generalized Production Rule System (GPRS), 91–109. New York: Springer-Verlag.

Findler, N. V.; Lo, R.; and Lo, C. 1987. A Pattern Search Technique for the Optimization Module of a Morph-Fitting Package. *Mathematics and Computers in Simulation* 29:41–50.

Findler, N. V. 1983. An Expert Subsystem Based on Generalized Production Rules. In Proceedings of the Sixteenth Hawaii International Conference on Systems Science, volume 1, 401–405. Honolulu: HICSS.

Findler, N. V., and Lo, R. 1983. A Note on the Functional Estimation of Values of Hidden Variables—An Extended Module for Expert Systems. *International Journal for Man-Machine Studies* 18:555–565.

Findler, N. V., and Morgado, E. 1983. Morph-Fitting—An Effective Technique of Approximation. *Journal of Mathematics and Computers in Simulation* 25:417–429.

Findler, N. V.; Brown, J. E.; Lo, R.; and You, H. Y. 1983. A Module to Estimate Numerical Values of Hidden Variables for Expert Systems. *International Journal for Man-Machine Studies* 18:323–335.

Findler, N. V. 1980. Pattern Recognition and Generalized Production Systems in Strategy Development. In Proceedings of the Fifth International Conference on Pattern Recognition, volume 1, 140–145. Los Alamitos, Calif.: IEEE Computer Society.

Causal Modeling System

Findler, N. V. 1990. Contributions to a Computer-Based Theory of Strategies—Chapter 7, Causal Modelling Systems (CMS and NEXUS), 187–239. New York: Springer-Verlag.

Findler, N. V., and Bozsahin, H. C. 1990. Representing Causal Knowledge. In Proceedings of the Fifth Rocky Mountain Conference on Artificial Intelligence, 332–337. Las Cruces, N.M.: Computing Research Laboratory.

A Predictive Man-Machine Environment

Findler, N. V. 1990. Contributions to a Computer-Based Theory of Strategies—Chapter 8, The Predictive Man-Machine Environment (PMME), 241-253. New York: Springer-Verlag.

Findler, N. V. 1988. A Prototype of a Man-Machine Environment for the Study of Air Traffic Control. In *Encyclopedia of Computer Science and Technology,* volume 19, eds. A. Kent and J. G. Williams, 3–21. New York: Marcel Dekker.

Findler, N. V.; Bickmore, T. W.; and Cromp, R. F. 1985. A General-Purpose Man-Machine Environment with Special Reference to Air Traffic Control. *International Journal for Man-Machine Studies* 23:587–603.

A System for Heuristic Retrieval of Information and Facts

Findler, N. V. 1982. Implications of Artificial Intelligence for Information Retrieval. In Proceedings of the ACM National Conference, 187. New York: Association of Computing Machinery.

Findler, N. V. 1979. A Heuristic Information Retrieval System Based on Associative Networks. In *Associative Networks—The Representation and Use of Knowledge in Computers*, ed. N. V. Findler, 305–326. New York: Academic.

Findler, N. V. 1979. Fuzzy Retrieval of Fuzzy Information. In *Knowledge Representation and Reasoning in the Humanities and Social Sciences*, ed. M. Borillo, 288–294. Rocquencourt, France: Institut de Recherche en Informatique et Automatique.

Computational Lexicometry

Findler, N. V., and S.-H. Lee. 1978. One More Step toward Computer Lexicometry.

American Journal of Computational Linguistics 15(2), Microfiche 43.

Findler, N. V., and Cipolaro, J. K. 1977. MARSHA, the Daughter of ELIZA—A Simple Program for Information Retrieval in Natural Language. *American Journal of Computational Linguistics* 14(4), Microfiche 66.

Findler, N. V., and Viil, H. 1974. A Few Steps toward Computer Lexicometry. *American Journal of Computational Linguistics* 1(1), Microfiche 4.

Findler, N. V. 1970. Some Conjectures in Computational Linguistics. *Linguistics* 64:5–9.

Distributed Air Traffic Control

Findler, N. V. 1987. Air Traffic Control: A Challenge for Artificial Intelligence. *AI Expert* 2:59–66.

Findler, N. V., and Lo, R. 1986. An Examination of Distributed Planning in the World of Air Traffic Control. *Journal of Distributed and Parallel Processing* 3:411–431.

Findler, N. V. 1985. The Role of Strategies in Air Traffic Control. In Proceedings of the Eighteenth Hawaii International Conference on System Sciences, volume 1, 522–529. Honolulu: HICSS.

Findler, N. V. 1984. Some Artificial Intelligence Contributions to Air Traffic Control. In Proceedings of the Fourth Jerusalem Conference on Information Technology, 470–475. Washington, D.C.: IEEE Computer Society.

Distributed Control of Nationwide Manufacturing Operations

Findler, N. V., and Reece, G. 1990. Toward a Blackboard Framework for Opportunistic Planning and Dynamic Scheduling in a Modular Job-Shop Scheduling System. In Proceedings of the IEA/AIE-90, Third International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 151–155. New York: Association for Computing Machinery.

Findler, N. V. 1989. Distributed Knowledge-Based Systems in Manufacturing. In Proceedings of the International Conference on Artificial Intelligence in Engineering, 17–33. Berlin: Springer-Verlag.

Findler, N. V., and Ge, Q. 1989. Perceiving and Planning before Acting—An Approach to Enhance Global Network Coherence. *International Journal of Intelligent Systems* 4:459–470.

Findler, N. V., and Ge, Q. 1988. Distributed Planning and Control for Manufacturing Operations. In Proceedings of AI'88, the Australian Joint AI Conference, 439–455. Sydney: ACS.

Findler, N. V., and Gao, J. 1987. Dynamic Hierarchical Control for Distributed Problem Solving. *Data and Knowledge Engineering* 2:285–301.

Distributed Control of Street Traffic Signals

Findler, N. V. 1991. Distributed Control of collaborating and Learning Expert Systems. In Proceedings of the IFAC International Symposium on Distributed Intelligent Systems. Washington, D.C.: IEEE Computer Society.

Findler, N. V., and Stapp, J. 1991. A Distributed Approach to Optimized Control of Street Traffic Signals. *Journal of Transportation Engineering.*

Intelligent Distributed Simulation Models

Sengupta, U.K.; Bozsahin, H. C.; Klesczewski, K.; Findler, N. V.; and Smith, J. W. 1990. PGMULES—An Approach to Supplementing Simulation Models with Knowledge-Based Planning Systems. Presented at the AAAI-90 Workshop on Simulation and AI, Boston, Mass.

An Excursion into Social and Cultural Anthropology by Artificial Intelligence—An Automated Dicovery System to Identify Rules for Inheritance, Succession, Marriage, Injunction against Incest, and Exogamy.

Findler, N. V. 1973. Kinship Structures Revisited. *Behavioral Science* 18:68–71.

Findler, N. V., and McKinzie, W. R. 1969. On a Computer Program That Generates and Queries Kinship Structures. *Behavioral Science* 14:334–343.

Nicholas V. Findler is currently research professor of computer science and director of the AI Lab at Arizona State University. He has worked in AI since 1957 (starting in Sydney, Australia). He has lectured in many parts of the world, published over 150 papers, and written or contributed to 30 books.

Uttam Sengupta has, at press time, just defended his Ph.D. dissertation, which deals with distributed planning and problem solving. He has an M.S. in physics from the Indian Institute of Technology and one in computer science from New Mexico State University. He has published several articles on both physics and AI.