DOCUMENT RESUME

ED 327 164 IR 014 789

AUTHOR	Abramson, Bruce
TITLE	Competent Systems: Effective, Efficient, Deliverable.
PUB DATE	90
NOTE	10p.
PUB TYPE	Reports - Research/Technical (143)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Artificial Intelligence; *Computer System Design; Design Requirements; Diagrams; *Expert Systems; *Models; User Needs (Information)

ABSTRACT

Recent developments in artificial intelligence and decision analysis suggest reassessing the approaches commonly taken to the design of knowledge-based systems. Competent systems are based on models known as influence diagrams, which graphically capture a domain's basic objects and their interrelationships. Among the benefits offered by influence diagrams is their underlying psychological and mathematical validity. For most users, the salient feature of influence diagram modeling is the precision and clarity that it forces on both the domain expert providing information and the system designer building the model. This paper presents a user-oriented perspective of a vertical approach to system design. It is stated that this design promises efficient development and rapid delivery of theoretically justified systems tailored to user need. (9 references) (DB)

Reproductions supplied by EDRS are the best that can be made from the original document.

U S DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it
☐ Minor changes have been made to improve reproduction quality
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

Competent Systems: Effective, Efficient, Deliverable

Bruce Abramson, Ph.D.
Assistant Professor of Computer Science
University of Southern California
Los Angeles, CA 90089-0782
(213) 743-8866

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Bruce Abramson
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ED327164

ABSTRACT

Recent developments in artificial intelligence and decision analysis suggest reassessing the approaches commonly taken to the design of knowledge-based systems. Competent systems are based on models known as *influence diagrams*, which graphically capture a domain's basic objects and their interrelationships. Among the benefits offered by influence diagrams is their underlying psychological and mathematical validity. Unlike standard rule- or frame-based systems, which are well motivated but poorly justified, competent systems *should* (according to an underlying theory) successfully meet a user's desires. For most users, however, theoretical justification is reassuring, but not crucial. To them, the most salient feature of influence diagram modeling is the precision and clarity that it forces on both the domain expert providing information and the system designer building the model. This paper presents a user-oriented perspective of our vertical approach to system design---one that promises efficient development and rapid delivery of theoretically justified systems tailored to user need.

Introduction

What is a "competent" system? Why should anyone be interested in systems that are merely competent when we already have "expert" systems? And what makes these systems so remarkably "effective, efficient, and deliverable?" The title of this paper was designed to be provocative; the history of intelligent (or knowledge-based or expert) systems has taught us to be wary of new catch phrases, new promises, new hype. In fact, it is precisely this type of skepticism that led our group, an interdisciplinary research team with backgrounds in artificial intelligence (AI), decision analysis (DA), psychology, decision support systems (DSS), financial forecasting, and medical information processing, to begin assessing the technology underlying knowledge-based systems.

This paper does not detail our work at the level that might be desired by an academic researcher in any one of these fields. Instead, it targets individuals and institutions armed with real tasks---diagnoses, forecasts, decisions, training exercises, etc.---that *should* be amenable to AI technology, but have yet to be successfully automated. We will provide the reader with a brief outline of our underlying assumptions, our techniques for designing systems, and our experiences with two real systems: Pathfinder, (which diagnoses diseases of the lymph system), and ARCO1, (which forecasts the price of crude oil)[1]. Readers should emerge convinced that our approach is widely applicable, effective, and efficient, that it promises to lead to rapidly deliverable systems, and that it warrants consideration for the tasks that they wish to automate.

[1] The author was primarily responsible for the design of ARCO1; all unpublished information is based on personal experience. Unpublished reports about Pathfinder are based on the author's conversations with its designers, David Heckerman, Eric Horvitz, and Bharat Nathwani.

IR014789

ERIC

Motivation

Through the late 1970's, expert systems were regarded as expensive, large-scale, experimental research projects; by the mid-1980's, they had become accessible to anyone owning a personal computer. The late 1980's witnessed reductions in R&D, as many initial boosters of the technology became disillusioned; poor system performance, budget overruns, and unacceptable returns on investment were common experiences. Both the lionization of expert systems and their subsequent fall from grace, however, were premature. Designers of early systems---like the pioneers of most new technologies---made some mistakes. A decade of experiences, however, provides the data necessary to sort the wheat from the chaff (or the baby from the bathwater, for the urbanites among us).

Stated succinctly, the designers of most existing expert systems attempted to capture the techniques employed by human experts, to model these techniques, and then to automate their models. Their elicitation (or knowledge acquisition) phases were usually guided by questions of the form, "what would you do in the following situation?," and responses were modeled as either production systems (large collections of IF. . .THEN. . . rules) or frames (descriptions of commonly encountered situations) (8). The resultant systems were expected to simulate expert human behavior.

Despite the obvious appeal of this approach, it lacks experiential, psychological, or mathematical justification. Experientially, it deviates both from the way in which expertise is attained and from the way in which devices are invented. First, people who set out to become experts in a narrow subspecialty rarely begin by focusing exclusively on their area of specific interest; they begin as broad-based novices or apprentices, narrow their focus as their training progresses and their competence increases, and eventually hone in on area of expertise. Experts, then, are simply the individuals with the highest degrees of competence among all people operating in a domain (hence the name competent systems). Second, few (if any) inventions have been based on mimicry; they usually exploit new technologies to address specific needs. Jets, for example, do not mimic birds, nor radar eyes. These inventions capture some of the characteristics of their natural counterparts, add a few unique features facilitated by their underlying technologies, and provide elegant solutions to important problems. Psychologically, the elicitation of procedural expertise is demonstrably inaccurate; people are notoriously poor at knowing what they know (6). Mathematically, production systems and frames both lack underlying formal theories (7). In short, systems oriented around mimicking human expertise were motivated more by a desire to see them work than by any reason to believe that they should work.

Competent Systems

Competent systems, and their underlying approach to system design, originated with our desire to design useful systems that are based on valid underlying theories and models (2). We are interested in developing a vertically integrated theory of system design---one that originates with the needs of a user community, captures information provided by an expert in a psychologically testable model, and is based on a formal and precise mathematical theory. Given our current target audience, the most relevant aspect of competent system design is the way in which it addresses user needs. The existence of validating psychological and mathematical theories, however, should reassure potential users and sponsors about the likelihood of a reasonable return on their investments.

Since most users interested in developing knowledge-based systems for their domain of expertise are in greater need of tools than they are of either colleagues or mentors, the design of a simulated expert is unnecessary as well as unrealistic. Systems should be designed to capture an understanding of a domain and its tasks rather than the behavior of its experts; *task analyses* must provide the first phase of system design. This simple idea-

--that knowledge-based systems should model domains and solve problems, rather than model experts and simulate behavior---forms the basis of the competent system design theory.

Core Problems

The first requirement of anyone wishing to become an expert is that he or she understand the domain, its objects, and the relationships among them. Novices and trainees must also begin by mastering commonly occurring tasks before they progress to rarer, more difficult, and potentially more important problems. Knowledge-based systems, like people, should enter a new domain at its most basic level. Only systems that have demonstrated an understanding of fundamental objects and a mastery of basic tasks should be allowed to progress to the next level. Thus, the initial stages of a task analysis should lead to the selection of an appropriate "core" problem for the domain. Although the *definition* of a core problem must remain rather loose, some of its general *characteristics* are enumerable. A core problem should be . . .

. . . well-defined and within the realm of human expertise.
. . . relevant to at least some of the people in the domain.
. . . just beyond the state-of-the-art.
. . . accompanied by a performance metric.
. . . the simplest problem that satisfies the above.

The adoption of a core problem corresponds to the strategy of selecting problems that appear to be relevant and solvable rather than those that look most exciting. Despite their relative simplicity, core problems are rarely trivial, as the case studies of Pathfinder and ARCO1 should demonstrate

Pathfinder, designed at Stanford University and USC (4), operates in the domain of hematopathology (diseases of the human lymph system). The first---and most obvious---question that a system designer could pose to an expert hematopathologist, is "How do I diagnose a disease of the lymph system?" The procedural orientation of this question, however, would lead to precisely the type of mimicry that we are trying to avoid. Thus, a better question would be "What information might I need to diagnose a disease of the lymph system?" Answers to this question are both within the realm of human expertise and relevant to many of the people operating in the domain. Nevertheless, simpler questions do exist: "What information might I need to differentiate between a given pair of diseases of the lymph system?" is obviously simpler and within the realm of human expertise, but unlikely to be relevant to anyone. The question "What information might I need to differentiate between each pair of diseases of the lymph system?" on the other hand, possesses all characteristics of a core problem. It is within the realm of human expertise, relevant to virtually everyone in the domain, and extremely simple. By iterating a seemingly trivial problem throughout the domain, Pathfinder's designers applied a *divide-and-conquer* strategy to knowledge-based system design, and thus eased both model construction and validation. In so doing, they also furthered the claim that resolution of their problem was just beyond the state-of-the-art, and facilitated the use of case histories with known diagnoses as a body of test data against which system performance could be measured.

ARCO1, designed at USC and the Atlantic Richfield Company (ARCO) (3), operates in a very different setting: the crude oil market. ARCO1 was commissioned by, and models the expertise of, members of ARCO's corporate planning group. Thus, the overriding question of interest is "How do I plan resource allocation for a major oil company?" Answers to this question are both procedural and extremely complex; although it might be a reasonable ultimate objective, it is a poor choice for the domain's first automated system. A good preliminary question, then, is "What is the most basic piece of

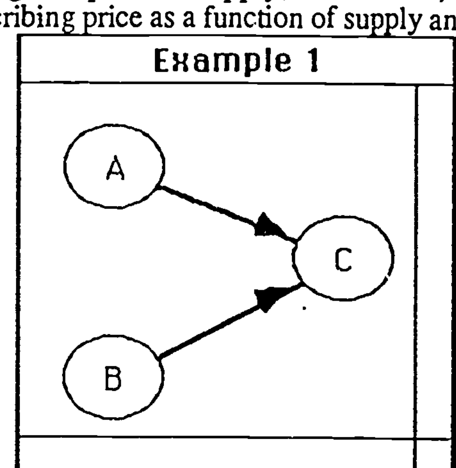
ERIC

non-trivial information needed for corporate planning?" The answer---a forecast of the price of oil---motivates an appropriate core question, "What information might I need to forecast oil prices?" Once again, answers to this question are within the realm of human expertise (at least to the extent that forecasting is tractable), they are obviously relevant to everyone in the domain, and the existence of forecasting tools indicates that stronger forecasts are just beyond the state-of-the-art. Although simpler questions may exist, none are obvious. This core question, like Pathfinder's, introduced some common design techniques into the realm of knowledge-based systems, (in this case, *subscripted variables*), that greatly eased the modeling and validation phases.

Influence Diagrams

The determination of an appropriate core problem is more a prerequisite for successful system design than a part of the actual design effort. The first true design phases---knowledge elicitation and formal modeling---must lead to an understanding of the domain's basic objects and of their *direct* interrelationships. The most straightforward representation of objects and relationships is neither a production rule nor a frame, but rather a graph.

In the graph shown below, A, B, and C each represent distinct objects, while the arcs from A to C and from B to C indicate that that the values of A and B each have some sort of influence on the value of C. In a medical domain, for example, A could represent the disease pneumonia, B the disease common cold, and C the symptom coughing. The arcs could then represent probabilities: the A to C arc indicates that pneumonia causes coughing with probability p, and the B to C arc that a cold causes coughing with probability q. In an economic setting, A might represent supply, B demand, C price, and the arcs an econometric formula describing price as a function of supply and demand.



This simple example masks a modeling technique of tremendous power and sophistication. Each object in the domain---and its relationships to the objects that influence it (and that it influences)---may be studied in relative isolation and modeled in its most natural and elegant form. The only restriction is that each node must contain a method for generating a single value (for the object that it models) for every combination of influences (i.e., sets of values assigned to the variables that point to it). In other words, any valid, fully specified, mathematical or probabilistic relation can be incorporated into the model. This flexibility stands in stark contrast to the relative uniformity required by most commercially available shells.

Graphical models of this sort have been studied under several names, most notably *influence diagrams* and *belief networks*. (Decision trees are a popular and widely used special case of these more general models). Mathematical and statistical analyses of influence diagrams have led to a variety of algorithms for tracking belief, propagating information, drawing inferences, simulating scenarios, and answering questions (7). Psychological studies have developed techniques for eliciting reliable and internally consistent sets of beliefs from experts, but only when these beliefs are represented as probabilities and other mathematical quantities (9). Thus, the models underlying competent systems can be justified along both mathematical and psychological dimensions---we know how to build good models, and we know how to manipulate the numbers within the models once they have been built (5). Furthermore, influence diagrams force domain experts to be precise about the assumptions that underly their analyses and to focus on direct relationships that are (generally) well understood; indirect relationships are implicit in the model, and can be calculated by functional composition. This degree of focus is crucial in domain modeling. The graphs underlying Pathfinder and ARCO1, shown below, are far too complex to be designed holistically. They each contain in the neighborhood of 150 different variables, equations, and conditional probabilities. Only careful decomposition of the domain into small groups of closely related objects made the modeling possible (1).

[INSERT THE PATHFINDER INFLUENCE DIAGRAM ABOUT HERE]

[INSERT THE ARCO1 INFLUENCE DIAGRAM ABOUT HERE]

Conclusions

This paper provided an overview of a new approach to the design of knowledge-based systems based on recent results from AI, DA, statistics, and psychology. The competent systems paradigm involves starting small and progressing through a series of increasingly complex problems. This approach promises efficient development and rapid delivery. Our experiences with Pathfinder and ARCO1 show that even systems restricted to core problems can be powerful and effective. Users in many domains should want to adopt our approach and models because their focus on simple, well-understood components of the domain address immediate needs, while the psychological and mathematical validity of their underlying models promises a high likelihood of success.

Competent systems share many characteristics with expert systems, yet differ from their conventional rule-based and frame-based counterparts in a few important areas: they stress the importance of incremental improvement, and they are based on precise, well-understood, formal models. Design principles, however, are just that: principles. The design of an actual influence diagram remains an art. Good design teams must possess expertise in both the domain being modeled and the modeling techniques being employed. Implicit in the availability of commercially marketed shells is that experts should be able to encode their own rule bases, model their own thoughts, and design their own systems. The sophistication and care necessary to model a domain as an influence diagram, however, emphasizes the need for a well-trained modeling expert. Influence diagrams must be viewed as the intellectual equivalent of industrial power tools; although anyone can use them, few will successfully build the systems that they desire, and many risk hurting themselves trying. Professionally constructed networks, on the other hand, promise to generate competent systems that are, in fact, effective, efficient, and deliverable.

References

(1) Abramson, B. *On Knowledge Representation in Belief Networks.* In Proceedings of the 3rd International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Paris, FRANCE, July 1990, pages 222-224.

(2) Abramson, B. & Edwards, W. *Competent Systems: A Prescriptive Approach to Knowledge-Based Advisors.* In Proceedings of the 1989 IEEE Conference on Systems, Man, and Cybernetics, Cambridge, MA, November 1989, pages 113-114.

(3) Abramson, B. & Finizza, A.J. *ARCO1: Extending the Frontiers of Knowledge-Based Technology.* University of Southern California technical report 90-10/CS, June 1990.

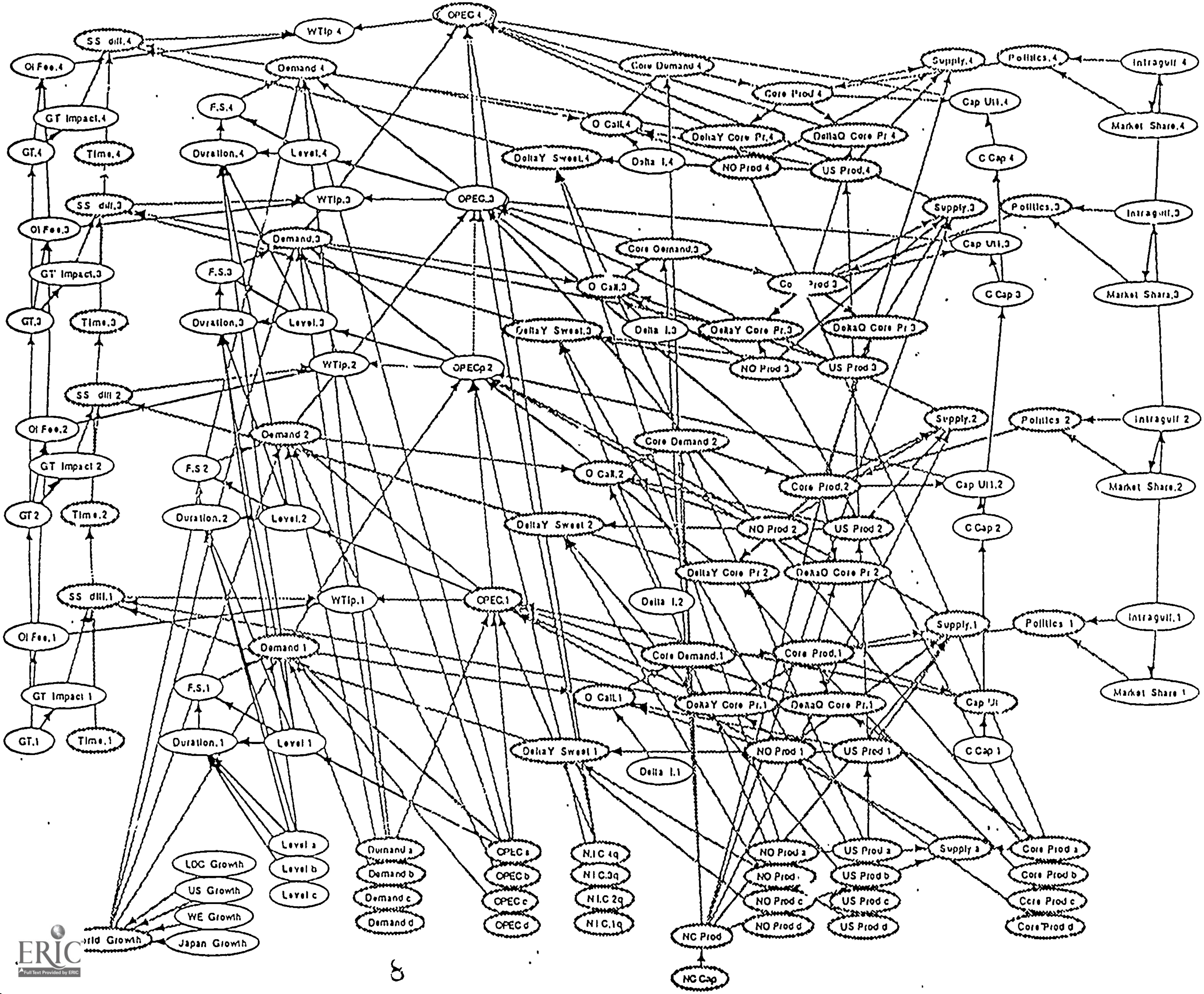
(4) Heckerman, D., Horvitz, E. & Nathwani, B. *Toward Normative Expert Systems: The Pathfinder Project*. Stanford University technical report KSL-90-08, 1990.
(5) Howard, R.A. & Matheson, J.E. *Influence Diagrams*. In R.A. Howard and J.E. Matheson eds., Readings on the Principles and Applications of Decision Analysis, vol II, pages 721-762, Strategic Decisions Group, 1984.
(6) Kahneman, D., Slovic, P. & Tversky, A. Judgement Under Uncertainty: Heuristics and Biases. Cambridge University Press, 1982.
(7) Pearl, J. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, 1988.
(8) Steels, L. *Components of Expertise*. AI Magazine, Summer 1990, 11(2):28-49.
(9) von Winterfeldt, D. & Edwards, W. Decision Analysis and Behavioral Research. Cambridge University Press, 1986.

About the Author

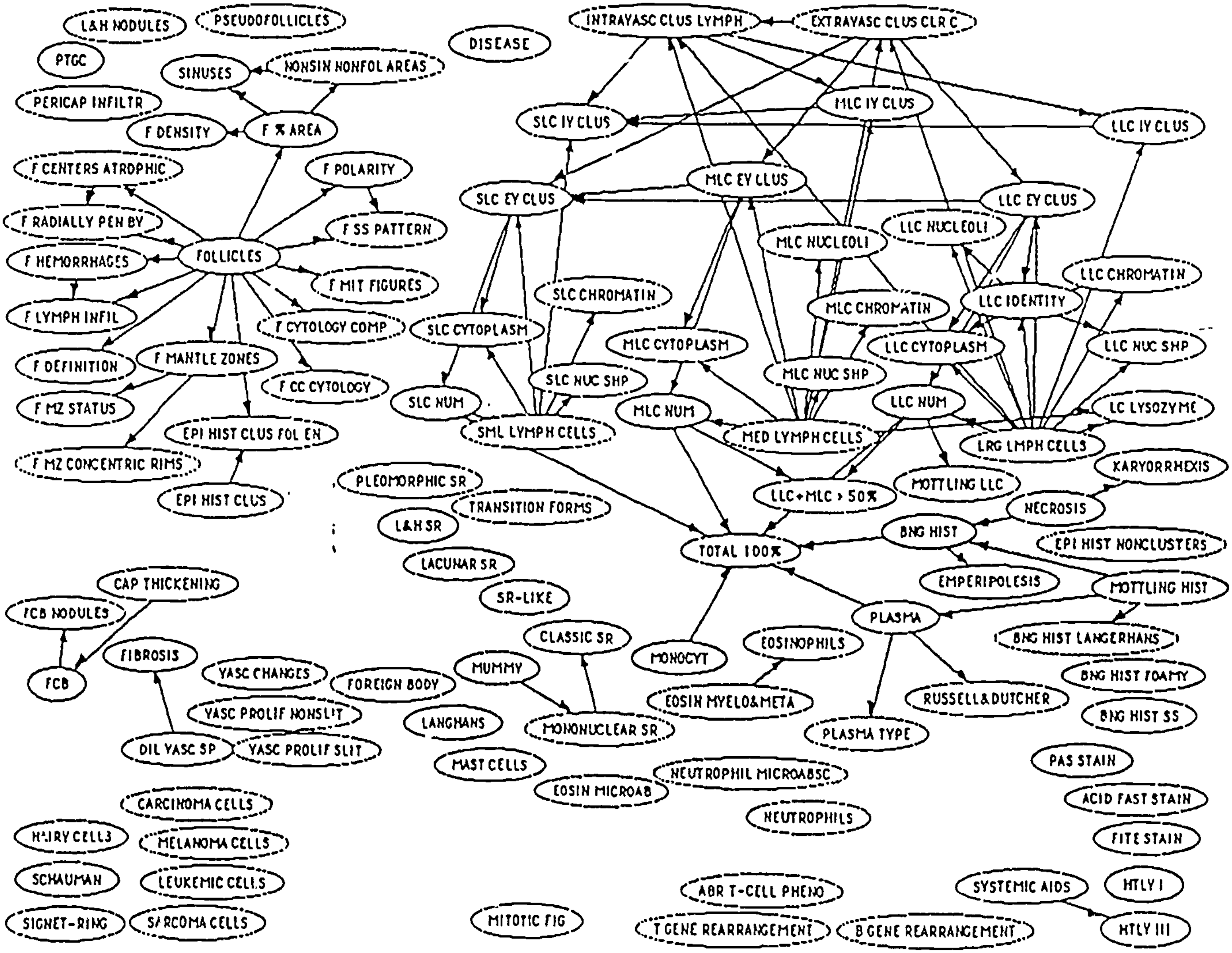
Dr. Bruce Abramson is an Assistant Professor of Computer Science at the University of Southern California (USC), Los Angeles, CA 90089-0782. He received a BA in 1983, an MS in 1985, and a PhD in Computer Science in 1987, all from Columbia University. While pursuing his doctorate, he spent two years as a researcher at UCLA. He has been at USC since 1987, where, in addition to his appointment in the computer science department, he maintains research affiliations with the Social Science Research Institute and with the Schools of Business and Accounting

Dr. Abramson's research focuses on the design and analysis of mathematically precise knowledge-based systems; he is particularly interested in the elicitation of knowledge from experts and its subsequent translation from the qualitative forms used by people to formal models that can be used by machines. As a result, his work tends to be highly interdisciplinary: His basic tools come from artificial intelligence, decision analysis, probability and statistics, and econometric modeling. The applications that he has studied (or is currently investigating) include two-player games, financial forecasting, robotics and automation, weather forecasting, and medical diagnosis.

Dr. Abramson is interested in learning about any potential application areas that may lead to consulting or research projects; he is particularly interested in economic and financial applications. He may be contacted at the above address, at (213) 743-8866, at (213) 745-0552, or via the arpanet at bda@cse.usc.edu.



The complete influence diagram underlying ARCO1.



The complete belief network for Pathfinder. The node DISEASE contains over 60 lymph-node diseases. The conditioning arcs from DISEASE to other nodes are not shown so that the conditional dependencies among features are highlighted.