Gaines. B. R. (1988). Applying expert systems. In Expert Systems: Strategies and Solutions in Manufacturing Design and Planning Kusialc.A.. ed. pp. 41-76. Society of Manufacturing Engineers. Dearborn, Michigan.

Applying Expert Systems

by Brian R. Gaines
University of Calgary

INTRODUCTION

The foundations of manufacturing are changing as automation and flexibility restructure the nature of the manufacturing process and its management. The advantages of the new technologies bring with them severe problems of coping with a complex and rapidly changing technological environment. Knowledge-based systems, through their capabilities to integrate heterogeneous knowledge processes, offer the possibility of management decision-support systems to aid the management of modern manufacturing. The technologies underlying knowledge-based systems are well advanced, and it is possible to develop major decision-support systems for manufacturing. The critical technologies to applications are the expert system support environment (ESSE) providing tools for interfacing with and validating the knowledge base, and knowledge acquisition tools for transferring knowledge from human experts to the knowledge base. This chapter analyzes the state of the art in knowledge-based systems and their applications in manufacturing, and illustrates the new tools for knowledge acquisition and transfer.

Major new opportunities have arisen in manufacturing, such as the following:

- Automation: New levels of maturity in automation have reduced the effect of unit labor costs and emphasized effective equipment investment;
  -- labor costs need no longer be critical competitive factors;
  -- it is possible to be competitive without relocating to remote territories;
  -- effective equipment investment is critical to a competitive position.
- Flexibility: Robotics and flexible automation enable short-run manufacturing and mass production to be comparable in unit costs;
  -- large production units can compete with small jobs for small-volume products;
  -- small job shops can compete with large production units for high-volume products.

With the new opportunities are associated new problems, such as the following:

- Economic change: The reduced sensitivity to labor costs and batch size changes the economic foundations of manufacturing;
  -- it moves the competitive edge elsewhere.

- New frameworks: Companies that only perceive the impact of the new technology on their existing businesses are likely to be deceived;
  -- the conceptual framework for manufacturing has changed.
- Critical factors: Operating factors that have been of secondary significance now become dominant;
  -- administrative overheads;
  -- inventory costs;
  -- effective communications with suppliers;
  -- effective communications with customers.
- Integration: This leads to an emphasis on integrated systems in which information flows are adequate to support the overall optimization of the manufacturing unit;
  -- new problem arise in the specification and design of integrated systems;
  -- major new problems arise in the management of integrated systems.

The effective management of the new technology has become of major importance, replacing people management as a primary concern. The flexibility, customization, and localization offered by integrated manufacturing systems are attractive but generate a new class of management problems.(1,2) The information flows required in manufacturing increase greatly with the diversity of product being handled and the rapidity of change between products. The knowledge required to manage these information flows is not readily available. Setting up a flexible production line requires different and higher level skills than those required to operate it once running. Additional skills are required to design a flexible manufacturing system (FMS) capable of handling a diversity of products.(3,4,5)

Management support systems are needed for the planning, implementation, configuring, and operation of integrated manufacturing systems.(6,7,8) Figure 1 is extracted from a paper entitled "Toward a Science of Manufacturing"(9) and shows the diversity of areas that have to be managed in a modern production unit. Each of the topics listed is one that merits specialist sessions at manufacturing conferences. Each topic has its own terminology, concepts, techniques, and specialists, yet consideration of all topics is essential to the effectiveness of the unit. We have created a situation in modern manufacturing where increasing in-depth technicality encourages more and more specialization, yet the principal requirements of plant design and operation demand generalists with a broad understanding of an increasing diversity of topics.

Fortunately, as often happens, the advances in technology that have created these problems also bring with them the means of solution. Modern information technology can be used to provide management decision-support systems that allow the generalist manager to cope with the diverse technicalities listed in Figure 1. Expert systems, in particular, provide a technology to cope with the diversity and complexity of the information involved.(10,11,12,13,14,15,16,17,18) However, an expert system alone cannot cope with the volume of information about different machine tools and robots, the computation required in grouping and the provision of numeric simulations for interactive planning and testing. An adequate support system for managing integrated manufacturing systems involves the

combination of expert system and simulation-based decision aids (19,20,21,22) with numeric computations, databases, communications, and, if it is to be used in operations, (23,24,25) instrumentation and control systems.(26,27)

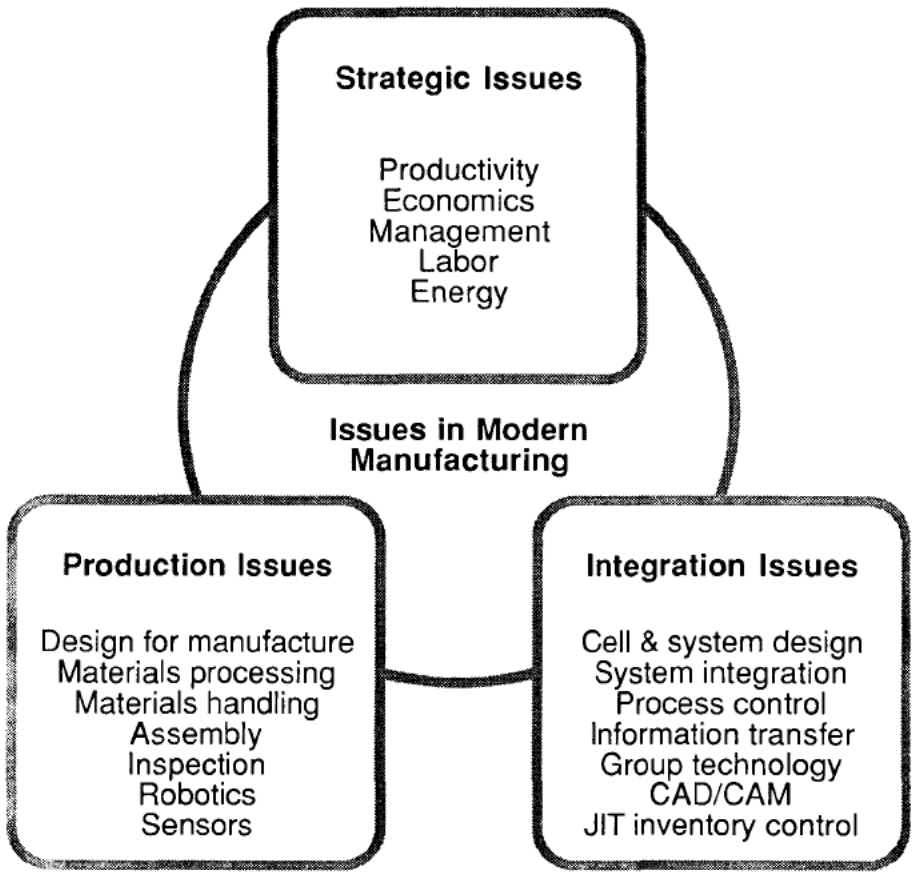

Figure 1. The diversity of issues in modern manufacturing.

In evaluating the application of expert systems in manufacturing, the same considerations apply as would for any new technology, as follows:

- Technology: What is the new technology, and how does it differ from existing comparable technologies?
- Maturity: What is the status of the new technology--experimental, developing, available, proven--do usable products exist?
- Applications: What areas of application are appropriate for the new technology?
- Usability: What is involved in using the new technology, and what will its impact be on existing personnel and organizational structure?
- Development: What is involved in developing applications using the new technology--skills, technical resources, costs, time scales, and likelihood of success?
- Cost effectiveness: What are the tradeoffs in applying the new technology--costs and impact in terms of competitive position?

The answers to all these questions must necessarily be tentative. The nature, time scales, applications, impact, and development of knowledge-based systems can be discussed in terms of our current knowledge of a new technology.(28,29) The validation of that knowledge and estimates of the cost effectiveness of the technology must await widespread experience of its use in a range of applications. Yet, the management of that applications experience requires estimates of the unknown parameters. This paper is concerned with provides such estimates in relation to the application of expert systems as part of management decision-support systems in modern manufacturing. It presents the essential features of knowledge-based systems, analyzes their role in relation to other information technologies, and discusses the particular problems of their application.

HETEROGENEOUS INTEGRATION THROUGH KNOWLEDGE-BASED SYSTEMS

What is it about knowledge-based systems technology that makes it better fitted to support the management of the diverse issues in modern manufacturing listed in Figure 1. The answer is best seen through contrast. A conventional corporate database might contain information relevant to most, if not all, of the areas outlined in Figure 1. It might contain powerful programs relating to each area. However, each of the programs and associated datasets would serve a specialist function within the corporation. There would be very little direct connection between them. They would be used in a connected fashion by the corporation, but the connections would be created by the human management of their application, not through information processing within the database itself.

Information technology to date has provided us with a heterogeneous collection of powerful, but disconnected, specialist tools. Knowledge-based systems provide the means of integrating these tools into a coherent knowledge support system aiding the management of complex modern technology. Human knowledge is based on heterogeneous integration on the pooling of diverse sources of information and diverse resources to achieve desired objectives. Knowledge-based systems provide a technology to support such heterogeneous integration within an information-processing facility, and hence also to support human objectives through those information-processing capabilities.

The essential features of a knowledge-based system are shown in Figure 2. Each specialist area of knowledge is represented as a knowledge island that has within it the datasets and procedures specific to that knowledge. Data from external sources triggers off processing within an island that eventually results in processed data being passed from one island to another. This in turn triggers further data processing in the other knowledge island that eventually propagates to other islands. Some islands have external connections that pass conclusions--recommendations for action--out to the external world. Islands are distinguished by the relative homogeneity of the data and processing within them. Connections between them represent data which is relevant to more than one island, but probably relevant in very different ways. The structure is fully recursive in that islands generally have sub-islands within them and group naturally into areas.

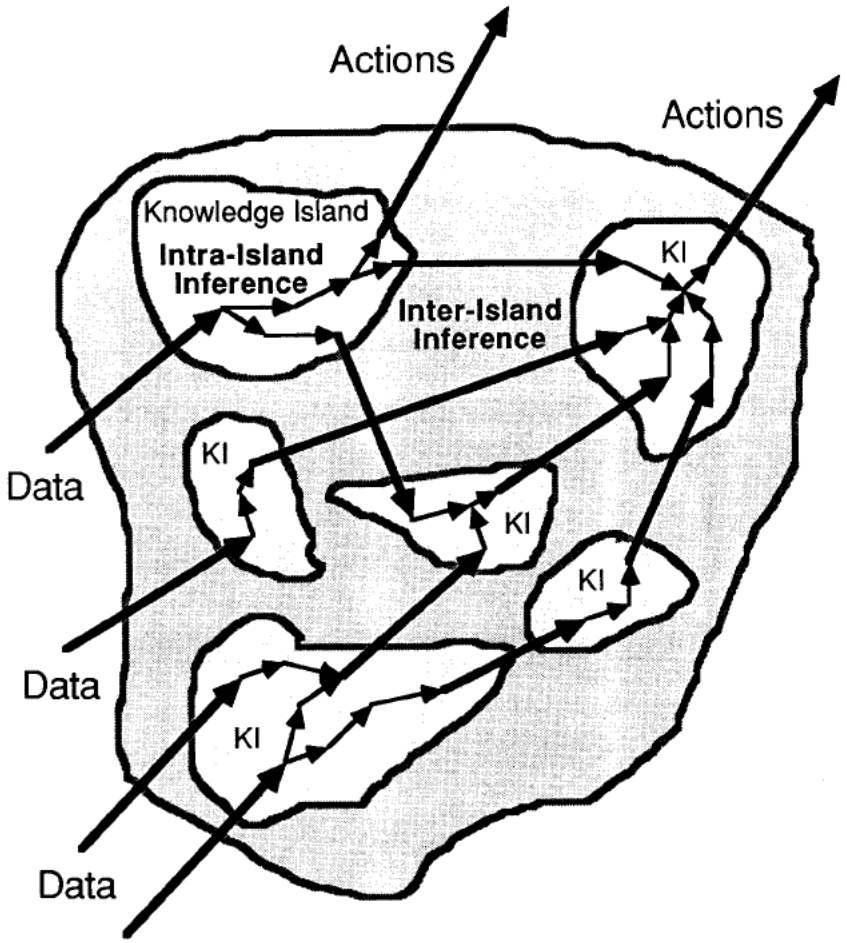

Figure 2. Inference processes within and between knowledge islands.

To make the abstract model of Figure 2 more concrete, Figure 3 shows an analysis of planning for flexible manufacturing in terms of the knowledge islands involved derived from interactive construct elicitation.(30,31) At the top left, the overall requirements specification is the starting point of any feasibility study of FMS. The key constructs here were found to be the obvious pair: low-volume/high-volume and low-variety/high-variety, but other significant constructs were also elicited that affect the feasibility of FMS implementation such as FMS already in use/FMS not in use and commitment to existing machines/new plant. The experience dimension and the difference between restructuring and greenfield situations are obviously relevant to the decision making, but could be missed in an analysis that concentrated only on quantitative parameters.

The experience/resources dimension refines naturally to the knowledge island concerned with overall resources and constraints. The volume/variety dimension refines to a quantitative description of the parts specification in parts per year and the number of different parts. Consideration of the resources and parts islands can give enough information to prime the overall system architecture island. This in its turn can lead to the fuzzy linguistic model approach.(32,33) The performance estimates and simulation

presentation form a basis for conclusions about the feasibility of an FMS for a particular requirement and the six knowledge islands involved may be grouped together as a feasibility study area. This subsystem is already a useful tool for the preliminary stages of FMS planning.

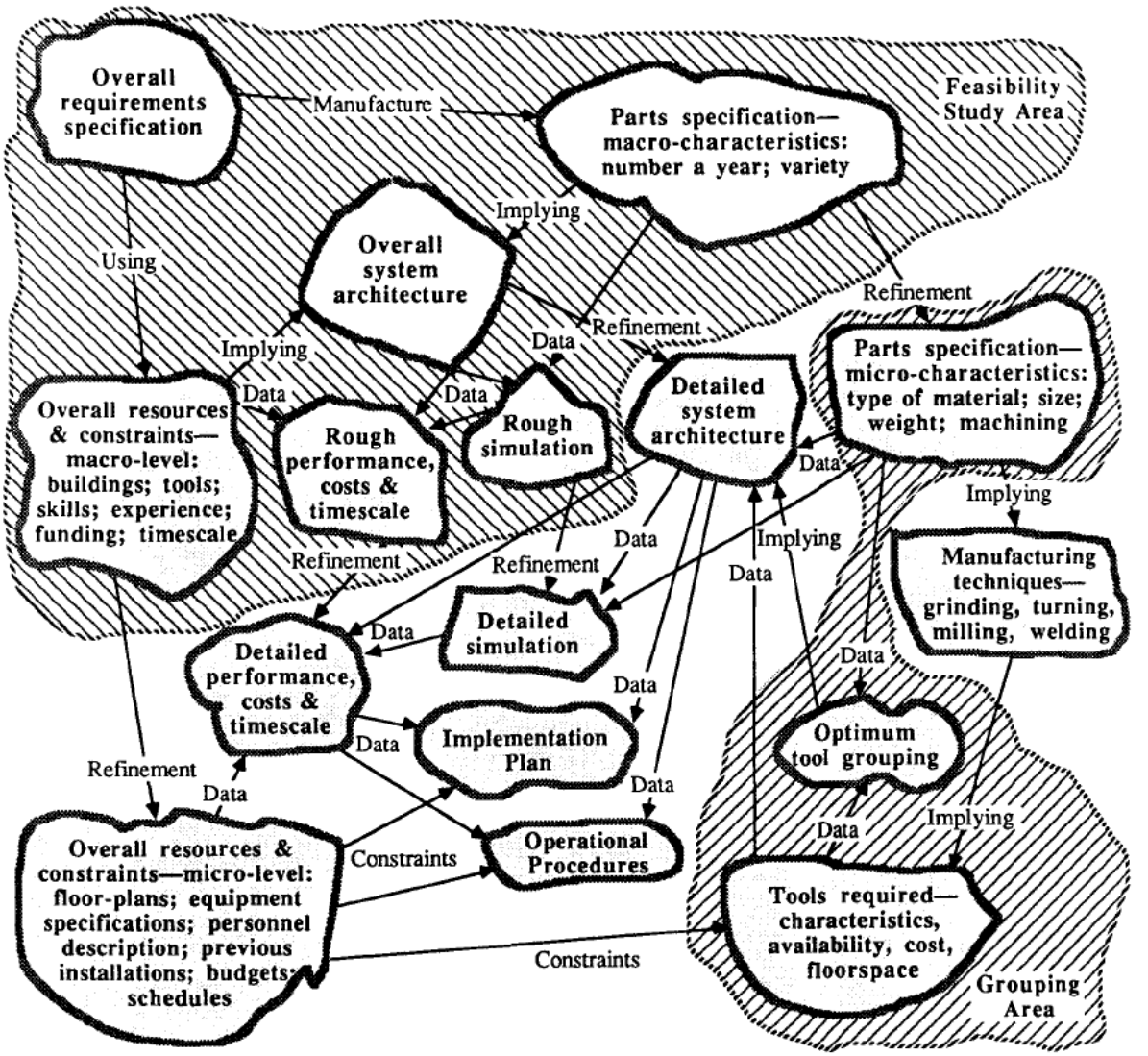

Figure 3. Knowledge islands in planning an FMS.

The knowledge islands described may be refined to a further level of detail on two main paths. The macro description of resources and constraints may be refined to a micro description that lists floor plans, personnel, machines, and so on. Much of this will be available in existing corporate administrative databases, but some information will probably be missing. The macro description of parts may be refined to a micro description that completely characterizes each part and its manufacturing process.(34,35) Much of this will be available in existing corporate engineering and production databases, but again some will be missing. The parts

specification leads naturally to a manufacturing techniques knowledge island that itself leads to that for tools required.(36) The parts and tools islands for another useful subsystem, the grouping area, where the optimum tool grouping knowledge island takes the detailed specification of parts and their machine utilization requirements and uses these to derive nearly optimal groupings of machinery.

The central region of detailed architecture, performance and simulation is complicated more by the data requirements than by the complexity of computation required. It is unlikely to be cost effective for any organization to gather the detail required on a wide range of machines and field experience (although as widespread use of MAP develops much more data is potentially available at low cost). The knowledge islands in this area will evolve through successive refinements of those already described in a way determined by mainstream industry trends and cost effectiveness. The tendency for leading suppliers tend to be used again because there is more knowledge of their products is one phenomenon of this region.

The importance of Figure 3 as an illustration of knowledge-based systems is the heterogeneity of the knowledge involved. The topic under consideration, managing FMS implementation, is itself relatively coherent, yet involves consideration of a diversity of requirements and constraints with little direct relationship between them. The whole of Figure 3 would itself be only one knowledge island in an overall manufacturing plan, which itself would be only one knowledge island in a corporate plan.

CONCEPTUAL STRUCTURES AND INFERENCE RULES IN KNOWLEDGE-BASED SYSTEMS

The technical foundations for the heterogeneous integration effected by knowledge-based systems is through their use of conceptual structures and inference rules rather than algorithmic data processing. These have proved to be powerful techniques for encoding computations that deal with a diversity of data and process different data domains and values in completely different ways. These are qualitative differences with no hard and fast boundaries--it is possible to recode much conventional data processing in this way; it also makes sense to incorporate algorithmic data processing in knowledge-based systems when it is natural to do so. The differences are best seen through examples.

One of the first knowledge-based system applications in an industrial context was the fuzzy logic control system developed in 1974 by Mamdani and Assilian. The system accepted human knowledge of control strategies verbally and encoded it directly as computer programs that acted on the environment.(37) This work was undertaken as part of a study of machine learning in process control, and the system controlled was a small steam engine. The verbal rules were of the form shown in Figure 4, part i. Priming the learning machine by such rules was first introduced because learning took too long. It was expected that the rules would act like verbal instructions to a human operator and aid the machine in setting up an initial approximate control strategy. However, it was found that rules such as this may be interpreted algorithmically to produce computer control programs that are more effective than those derived from classical control theory. Additionally, the learning machine then proceeded only to "learn"

less effective strategies. Hence, interest switched to the process whereby human expression of verbal rules that appear vague can lead to highly effective control strategies, that is to the development of expert systems.

The rules on which a particular expert system is based may be seen as ways of representing knowledge. This is not a new concept as such rules have been used in the past quite independently of computer systems. Figure 4, part ii, is an excerpt from a training manual for lime kiln operators.(38) It seems that the encoding of expertise into a set of fairly vague but highly applicable rules is a standard way of making it available to others. These rules generate a highly effective controller for a kiln when encoded in a computer, whereas conventional process modeling using an algorithmic approach is ineffective.(39) The reason for the success of the rule-based approach to such control problems is that the physical system being controlled is highly discontinuous in its phase space. Numerical control algorithms generally assume some form of continuity, but rule-based systems can encode control procedures that have very different actions in neighboring cells in the phase space. This is an example of heterogeneous integration at the micro level of the system. A similar model applies at the macro level, but concepts such as phase space are usually less useful for systems with large numbers of asynchronous loosely coupled components.

If the pressure error is positive and big and the change in pressure error is not negative medium or big

Then make the heat change negative and big

(i) Rule from a fuzzy logic controller

1. When the burning zone temperature is drastically low:
   (a) reduce kiln speed;
   (b) reduce fuel.

2. When the burning zone temperature is slightly low:
   (a) increase I.D. fan speed;
   (b) increase fuel rate.

(ii) Rules from a lime kiln operator's manual

Figure 4. Examples of expert system inference rules.

Rules are a natural way of expressing relations between diverse data structures. A simple numerical relation can be embedded in a rule, for example:

If x has any value, then y=2*x

and more complex, discontinuous relationships can be expressed in sets of rules, for example:

If x>7,	then y=2*x
If -2<x≤7,	then y=3*x
If -2≤x,	then y=x

Nonnumeric relations are as readily encoded, for example:

If machine not numerically controlled, then machine requires operator

However, rules become much more powerful if combined with conceptual structures that allow knowledge to be expressed in the most abstract and general form, and then instantiated in specific cases. The specific cases can inherit most of the properties of the general case, overriding some and adding others as necessary. Rules that apply at the abstract level can encompass specific instances of those rules applied to specific instances.

Figure 5 shows part of the conceptual structure for Fox's (40) knowledge-based job shop scheduling system.

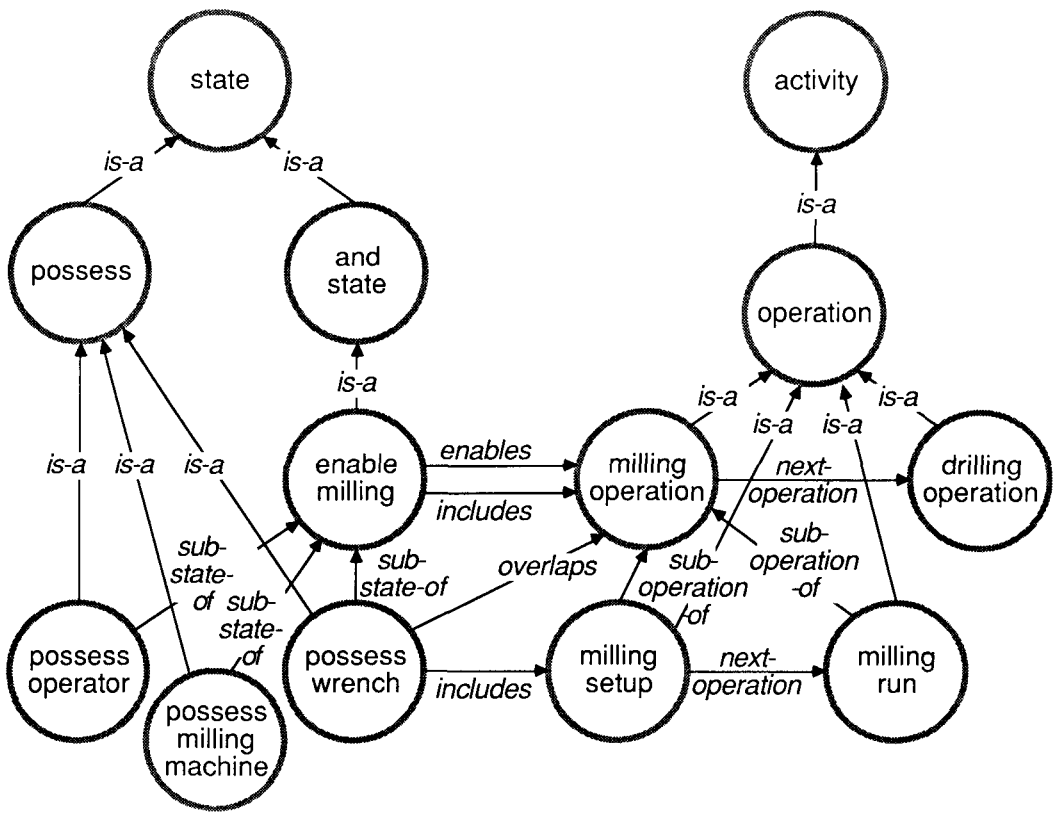

Figure 5. A conceptual structure from a job shop scheduling system.

The "is-a" relation expresses the relation that something is a more specific instance of a more abstract concept, for example, that to "possess" something is a "state," so that rules that apply to relations involving states will apply to instances involving possession. The other relations shown in this figure are largely concerned with the temporal relations involved in machining, including preconditions such as "enables," and sequencing such as "next operation."

The combination of conceptual structure, associated inference rules, and specific instances of concepts (or "facts") has come to be called a knowledge base, as a generalization of the term "database." Indeed a conventional database may be regarded as a special instance of a knowledge base in which the fields, their names and relations are the conceptual structure, and normalization requirements are inference rules that are applied at update time rather than at access time. It is the capability to apply inference rules at access time that most distinguishes databases from knowledge bases. In a database, all relations must be expressed extensionally, that is by a data object satisfying the relation being present. Whereas, in a knowledge base, relations can be expressed intensionally, as constraints used to generate data objects at access time. The added power of intensional expressions can be seen if one considers the storage requirements of storing the positive integers in a database--in a knowledge base, the rule that something is an integer if its value is one more than an integer, together with unity as an instance, is sufficient.

APPLICATION OF KNOWLEDGE-BASED SYSTEMS

The previous sections have emphasized the abstract concept that knowledge-based systems provide an effective means of providing management support systems for situations requiring the integration of heterogeneous sources and forms of knowledge. Hayes-Roth (41) has summarized a wide range of experience of applications of knowledge-based systems in more concrete terms.

He provides the following examples of situations appropriate to knowledge-based systems:

- The organization requires more skilled people than it can recruit or retain
- Problems arise that require almost innumerable possibilities to be considered
- Job excellence requires a scope of knowledge exceeding reasonable demands on human training and continuing education
- Problem solving requires several people because no single person has the required expertise
- The company's inability to apply its existing knowledge effectively now causes management to work around basic problems.

He provides the following examples of product innovations now possible:

- Active instruments that solve problems by producing answers instead of data
- Systems that explain how they work and how to use them

- Corporate memories
- Reasoning aids and thinking prosthetics
- Accident-proof machines
- Automated sellers
- Hypothesis and expectation management systems.

He presents the critical issues as follows:

- The power is in the knowledge
- Mining, molding, assembling and refining expert knowledge are the main problems
- Familiarity with appropriate software tools can drastically reduce costs and speed development
- Managers must grow to harness this new technology; it's not business as usual
- This is the area of maximum growth potential for computer applications.

These are still fairly abstract guidelines and an actual application is valuable in illustrating some of the concepts and issues involved. Figure 6 shows the expert system shell Nexpert (42) operating on a Macintosh computer in an industrial control situation.

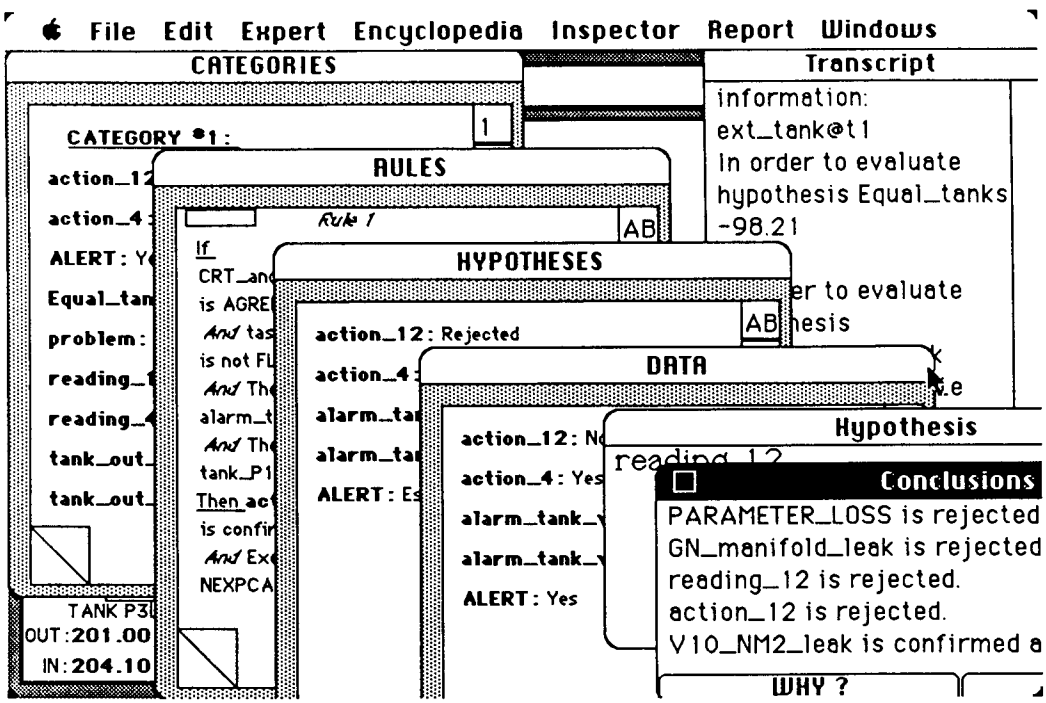

Figure 6. A modern multiwindow, menu, and icon environment in Nexpert.

The screen configuration shows the following:

- Transcript window keeps a log of the complete interaction, data entered, hypotheses entered, inferences made, conclusions reached, and so on

- Categories window shows the conceptual structure of the rules and data
- Rules window shows the rules in the knowledge base
- Hypotheses window shows the hypotheses entered
- Data window shows the data entered or inferred
- Hypothesis window shows the current hypothesis under consideration
- Conclusions window shows the conclusions reached during inferencing
- Prompt window (a one-line strip at the bottom of the screen, just visible in Figure 8) shows what action would take place if the mouse were to be clicked at the current cursor position.

Figure 7 shows a further window, the rule editor, which illustrates some of the care taken about usability in Nexpert's design, as follows:

- The use of the four "head" icons at the bottom to indicate how to add a rule, modify a rule, make a copy of a rule, and delete a rule, respectively
- The use of the "writing" icon at the bottom right to indicate how the rules may be written to disk
- The use of the "page turning" icon at the lower left of the window to indicate how the next rule may be accessed
- The use of the "alphabetic index" icons on the right to indicate how the rules may be accessed by the first letter of their conclusion.

Note also how the rule itself is automatically formatted for clarity in presentation. The prompt window is particularly helpful for initial users who are not familiar with the icons because moving the cursor to an icon brings up an explanation of the action involved as a prompt.

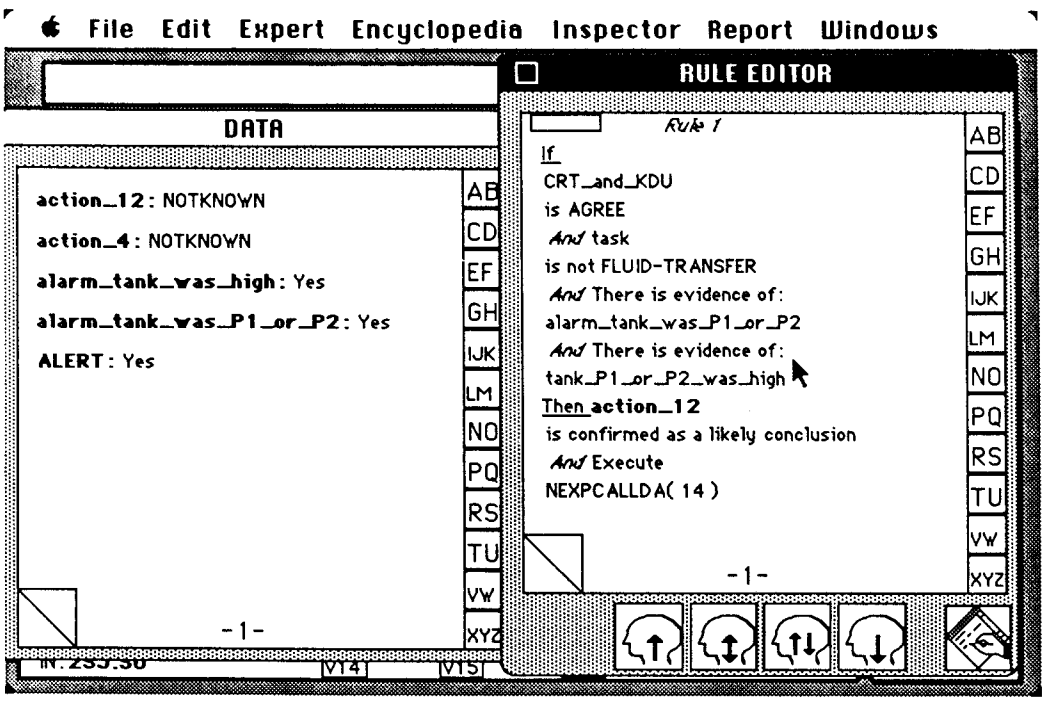

Figure 7. The rule editor dialog environment.

The graphic dialog capabilities used in Nexpert are important not only in their standard window, menu, and icon forms, but also in their capability to allow mimic diagrams to be used in highly customized user interfaces. Figure 8 shows a mimic diagram monitoring the status of a chemical plant through data received from instruments communicating through a Macintosh serial port. The operator can see the status of the plant through the numeric indicators and also actuate the plant by clicking on the valve symbols causing them to open and close.

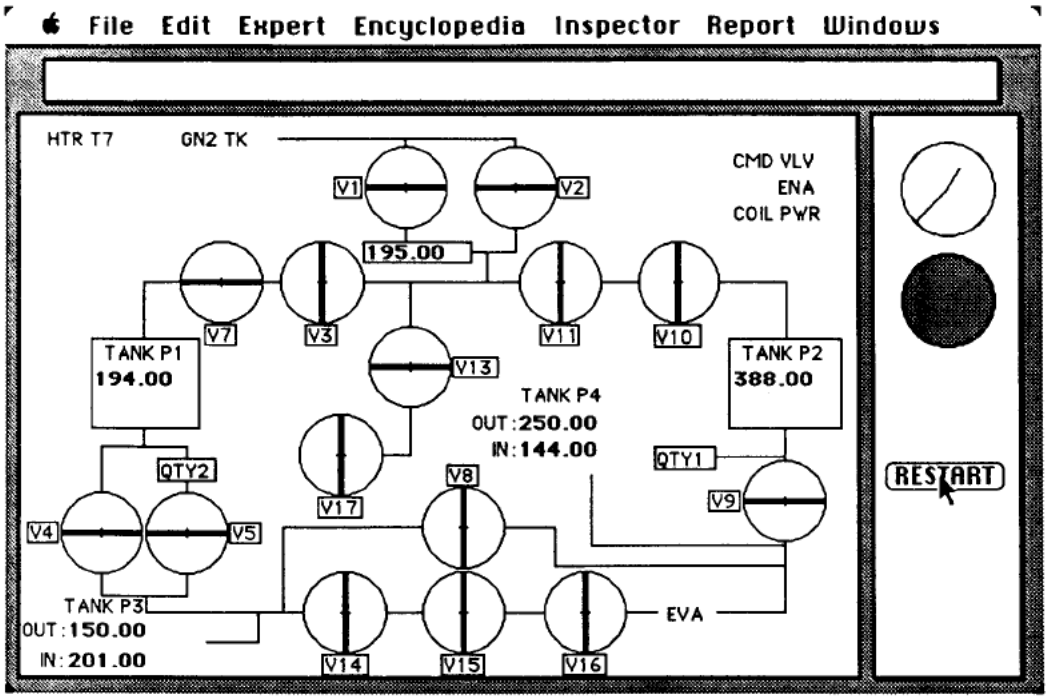

Figure 8. Process control and diagnosis through a mimic diagram of gages and semaphores.

An expert system is active in Nexpert while the mimic diagram is in operation. It monitors the status of the plant, and its rules are activated when values go outside safety limits. Figure 9 shows a dialog with this system triggered off by an alarm condition. It is asking for the task to be performed currently. The operator clicked on the "options" button when sure of possible responses. This causes them to be tabulated, and the operator has selected "FLUID-TRANSFER" from the table. Note that this list is generated automatically by the expert system shell by checking the rules that might use the data to be entered and determining what values they test for in their premises. Thus, an automatic help facility is generated that is bound to be meaningful in terms of the actual behavior of the system. This reduces the load of dialog design and ensures that the help facility is always in step with the actual knowledge base.

The operator may request an explanation of why the data is required by clicking on the "why?" button in Figure 9.

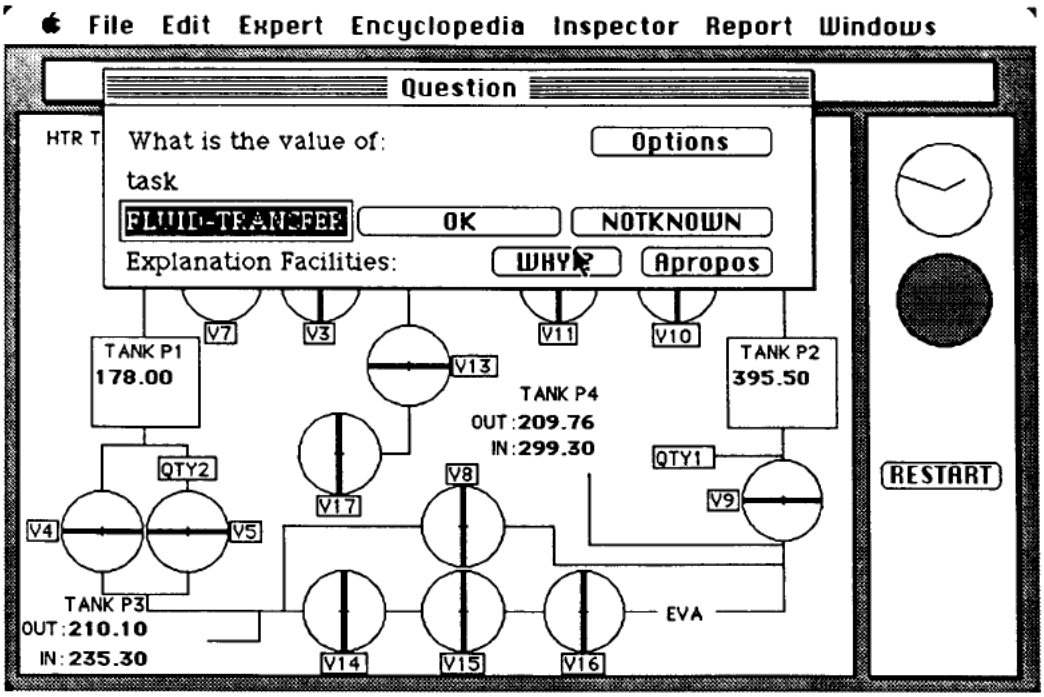

Figure 9. A consultation dialog with a plant fault detection expert system invoked by abnormal data.

An explanation is given in terms of the rule being tested that requires the data as shown in Figure 10. The explanation may be further expanded as required.

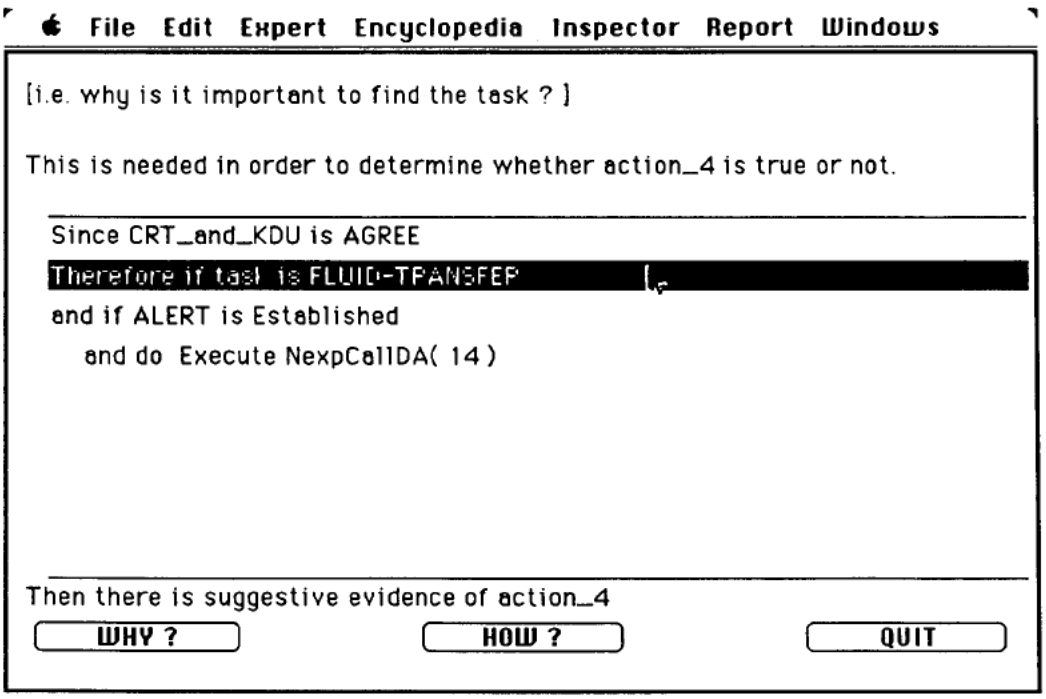

Figure 10. Explanation facilities.

Clicking on the second premise in Figure 10 highlights it, and then clicking on the "why" button again produces a broader explanation of why the task has to be known in terms of all the rules that use its value in their premises. Clicking on the "how" button at this stage queries in what ways the value of the task may be determined. Such explanation facilities are an important feature of any knowledge-based system.

When the operator has entered the data required, the expert system recommends actions that will result in containing the problem that has arisen. A second expert system is then invoked that diagnoses the likely source of the problem and indicates it on the mimic diagram as shown in Figure 11.

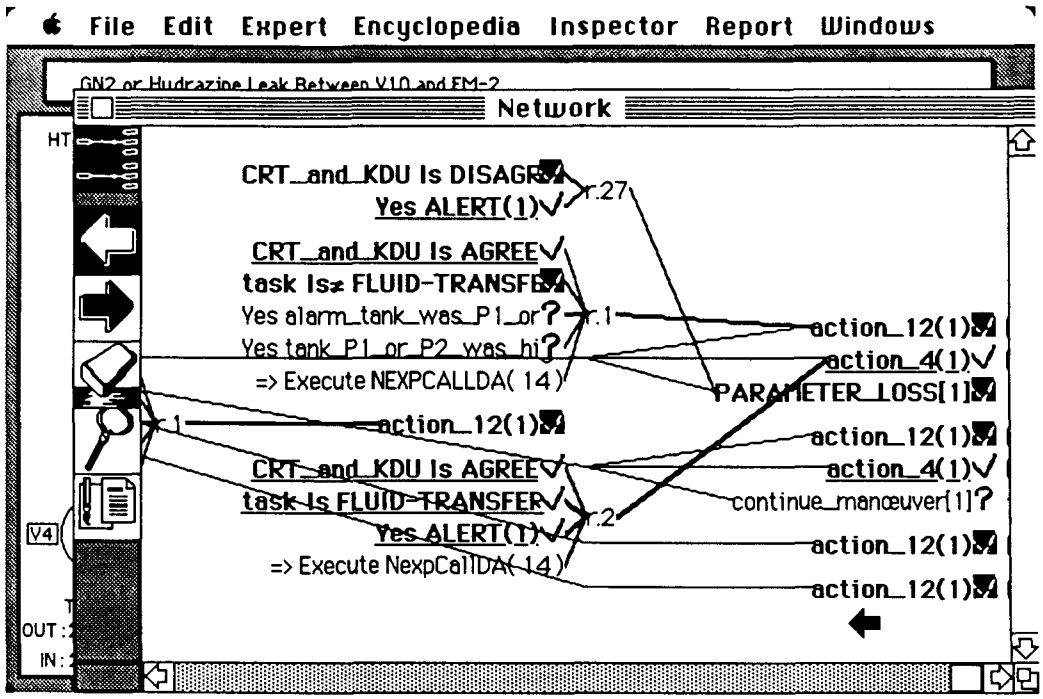

Figure 11. Mimic diagram.

Nexpert also uses the graphic capabilities to show the data and inferences in graphic form as shown in Figure 12. The user can navigate the knowledge base during an interaction and determine which rules have been invoked, and which rules are capable of being invoked, based on the data entered through the instruments and by the operator.

This example shows many of the features of knowledge-based system applications, the underlying knowledge structures, the representation of knowledge in a natural way for the user, and the capabilities to support intelligent dialog and to give explanations.

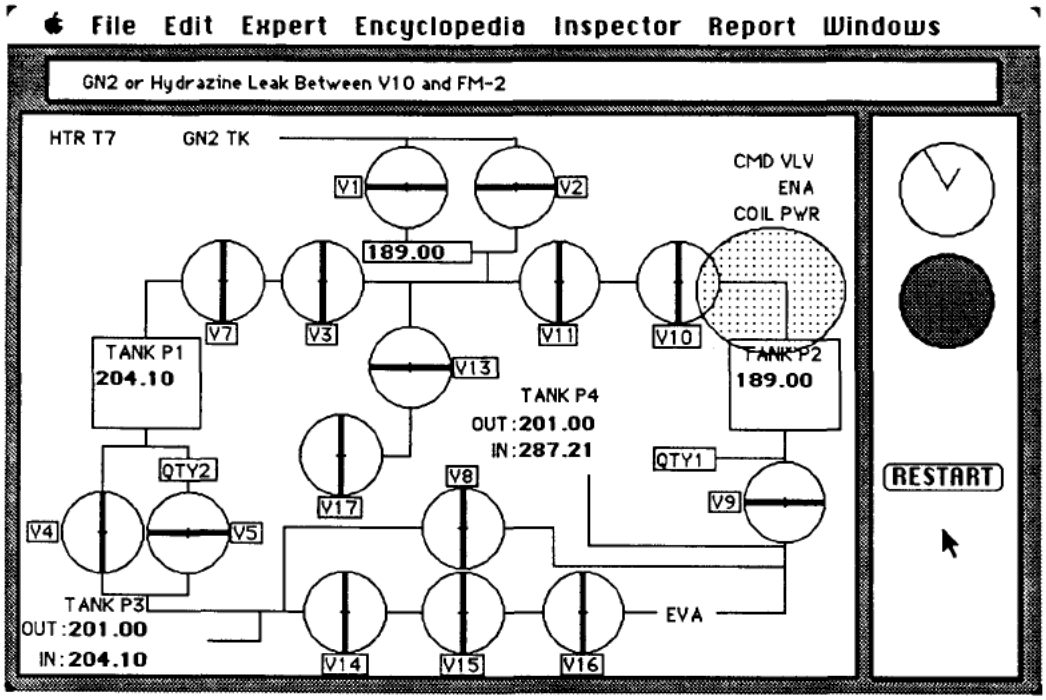

Figure 12. Graphical representation of the relations between data, rules, and inferences during a consultation.

THE ARCHITECTURE OF KNOWLEDGE-BASED SYSTEMS

One problem that managers often face in applying knowledge-based systems is the sheer variety of technologies apparently involved--should they use Lisp machines?--should they train their staff in Lisp or Prolog?--how does an expert system shell compare with existing programming languages?--what is knowledge engineering and how does it differ from systems analysis? The problem is that knowledge-based systems are at the top of the virtual machine hierarchy that has developed in computing, and there are many layers of technology between an expert system and the computer on which it is running. In theory, system developers should not need to know about the lower levels of the hierarchy--machine architectures, operating systems and implementation languages are remote from knowledge processing. However, in practice, these lower levels are the foundations on which systems are built and any defects in them can undermine the functionality of the upper levels.

Figure 13 shows significant levels of the virtual machine hierarchy for knowledge-based systems, as follows:

- Knowledge acquisition tools: At the top of the hierarchy are the tools for automating the systems analysis for expert systems, through automatic interview procedures, modeling actual expert behavior, and analysis of knowledge in textual form. Automation of systems analysis is more significant for expert systems than for conventional data processing because they tend to be targetted on narrow areas of in-depth specialist knowledge, rather than broad general problems.

In conventional terms, expert systems are satisfying the requirement for highly customized data processing based on idiosyncratic operations rather than general algorithms.

- Expert system support environment (ESSE): At the second level of the hierarchy is the equivalent of the application programming support environment (APSE) in conventional systems. The ESSE will be described in more detail in the next section.

Figure 13. The virtual hierarchy for knowledge-based systems.

- Expert system shell: At the third level of the hierarchy is the expert system shell as a run-time environment that elicits problem-specific information from the user, provides advice based on its knowledge base, and explains that advice in as much detail as required.
- Expert system shell development language: At the fourth level of the hierarchy is the language in which the expert system shell is written, generally a special-purpose environment for coping with knowledge representation and inference.
- Implementation language: At the fifth level of the hierarchy is the implementation language that actually interfaces to the computer. This tended to be Lisp in the early days of expert systems, but as speed and space efficiency have become significant, and knowledge representation has become better understood, other languages that support dynamic data structures such as C and Pascal have become widely used.
- Operating system: At the sixth level of the hierarchy is the operating system within which the implementation runs. This needs to provide good interfaces to other programs, large databases, and communications.
- Machine architecture: At the lowest level of the hierarchy is the machine on which the expert system runs. This is really irrelevant, but the use of Lisp in the early days of expert systems has focused attention on Lisp machines. Nowadays any good workstation is adequate.

The ESSE and knowledge acquisition tools shown in Figure 13 are the most critical requirements for knowledge-based system applications. Figure 14 shows their roles in the basic architecture of an expert system in relation to the roles of the users involved, as follows:

- Knowledge base: The central ring shows the knowledge base of conceptual structures, facts, and inference rules that is the heart of an expert system.
- Expert system shell: The middle ring shows the expert system shell that provides computational facilities for applying the knowledge base to user decision support.
  - -- The application system at the bottom tests the user's hypotheses or seeks to satisfy the goals by deriving the consequences of facts about a particular situation reported by the user, using the general facts and inference rules supplied by the expert and edited by the knowledge engineer.
  - -- The explanation system to the lower left answers queries about the way in which facts have been derived in terms of what information and inference rules have been used.
  - -- The acquisition system to the upper left provides tools for interviewing the expert to obtain the vocabulary, and general facts and inference procedures in terms of it.
  - -- The display system at the top provides tools for presenting the knowledge base in an understandable form, and the relations between the facts and inference procedures it contains.

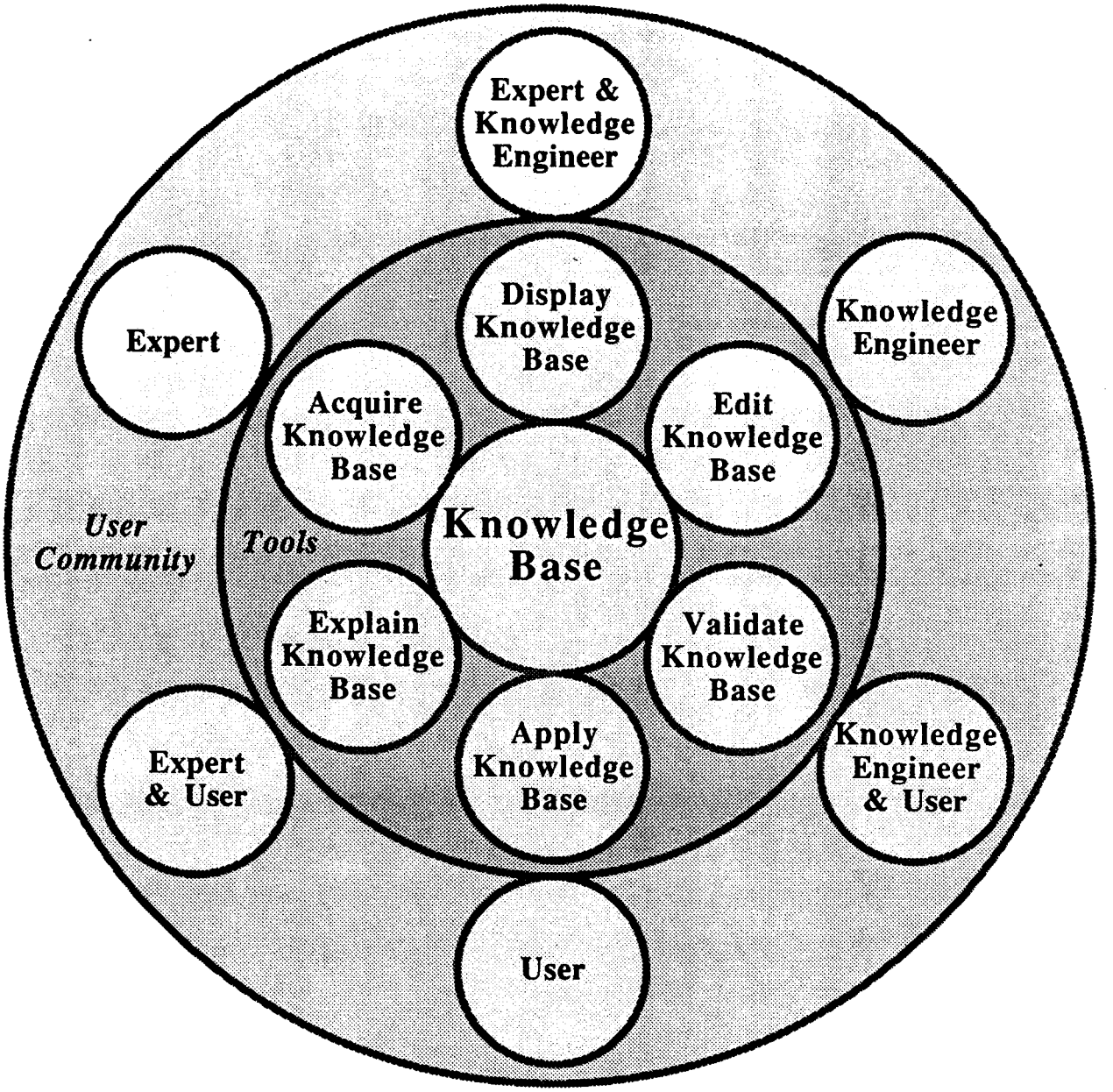

Figure 14. The architecture of a knowledge-based system in relation to its operators.

-- The edit system to the upper right provides tools for editing the knowledge base while maintaining its integrity in terms of the vocabulary, variables, and operations used.
-- The validation system to the lower right provides tools for checking the knowledge base against specific case histories with known consequences.

- Operators: The outer ring shows the three classes of operator of expert systems.
  -- Users come to the system for advice on specific problems in a particular domain, supplying it with specific facts and their hypotheses about consequences or goals.

-- Experts come to the system to transfer their knowledge about the problem domain, supplying it with generally applicable facts and inference procedures.
-- Knowledge engineers act as intermediaries between the expert and the system, helping to encode knowledge and validate the operation of the resultant expert system.

In considering system development, it is important to note that these operator roles may be combined within a single individual, such as the "expert and knowledge engineer," making it dangerous to assume that three self-contained working environments can cater for the different levels of skill and conceptual frameworks involved. It is better to think in terms of a single integrated environment that allows different levels of access through mutually consistent operating procedures. Most current expert systems do not provide such an integrated environment, and complete modules are missing or inadequately implemented, but Figure 1 provides a framework within which all existing systems can be embedded.

Most current expert system shells do the following:

- Offer a limited knowledge base represented in terms of production rules and frames
- Have weak application capabilities limited to deriving consequences and testing assertions
- Give simple explanations in terms of the facts and rules used
- Have no integrated acquisition capabilities
- Have limited capabilities to display the knowledge base, sometimes in graphical form
- Have a user interface to the knowledge base editor that requires understanding of internal data structures
- Have limited validation capabilities and no integrated database of case histories.

However, rapid improvements are taking place in the knowledge representation, inferencing, explanation, display, and editing capabilities of expert systems.(43,44,45,46,47) Most current knowledge acquisition is done by manual knowledge elicitation places too current knowledge acquisition is done by manual knowledge elicitation that places too much load on the knowledge engineer and inadequately involves the expert. However, automatic tools for rapid prototyping are becoming increasingly important (48,49,50) and are described in the next section.

KNOWLEDGE ACQUISITION FOR KNOWLEDGE-BASED SYSTEMS

The knowledge acquisition tools in Figures 13 and 14 are the most radical departure from classical adaptation development through system analysis. The conventional approach in decision and control system design is the instrumentation, data collection, modeling, and optimization sequence shown on the left of the diagram in Figure 15. This approach to system design underlies the methodologies of the physical sciences and technologies based on them. It has the merit that it has been extremely successful in engineering much of the technological infrastructure of our current society including our manufacturing industries.

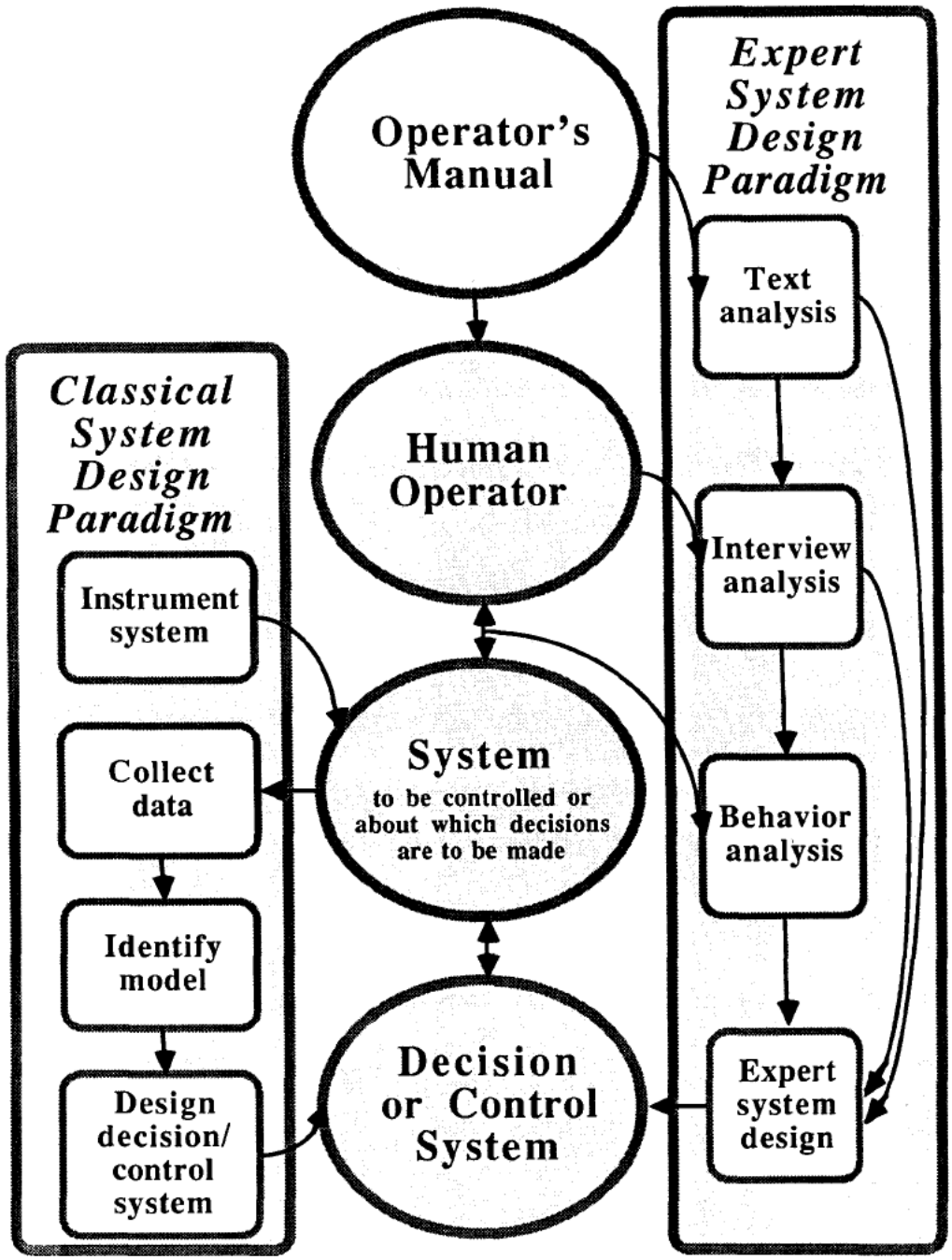

Figure 15. Classical and expert system design paradigms.

However, the approach shown in Figure 15 is successful only to the extent that the systems under consideration are amenable to instrumentation and modeling. Its greatest successes have been where this amenability can be achieved normatively, that is in cases where the system to be controlled is itself a human artifact. For example, linear system theory has become a major tool in system engineering not because most systems are linear--they are not. The implication is in the opposite direction: that linear systems are mathematically tractable and that we design systems to be linear so that

we may model them readily. The discontinuous control rules of Figure 4 are effective when linear system theory fails with plant such as a lime kiln where we cannot design the basic process to fit our mathematical models.

However, the control rules of Figure 4 were generated by, and are meaningful to, human operators, and the right hand side of Figure 15 shows the classical system paradigm extended to include the derivation of design information from the human operator. The obvious approach is interview analysis, and this has been used in many studies. However, often the operator is unable to give a verbal description of his skill, or suggests an incorrect basis for it, and behavior analysis through computer modeling of his input-output behavior gives better results.(51) Text analysis of the operator's handbook is another significant source of information about intended behavior.

A number of tools have been developed to automate the knowledge-based system design paradigm shown on the right of Figure 8 by interactively conducting a structured interview with experts, eliciting the experts' behaviors and then inductively modeling them, or by analyzing textual material. No one approach is adequate to capture all aspects of expertise, and recent systems combine all three techniques. The next generation of knowledge acquisition tools can be expected to integrate the development processes of both sides of this diagram.

The general requirements for a system to automate knowledge acquisition for knowledge-based systems are as follows:

- The knowledge acquisition system tools should be domain independent
- The tools should be directly applicable by experts without intermediaries
- The tools should be able to access a diversity of knowledge sources including text, interviews with experts, and observations of expert behavior
- The knowledge acquisition system should be able to encompass a diversity of perspectives, including partial or contradictory input from different experts
- The system should be able to encompass a diversity of forms of knowledge and relationships between knowledge
- The system should be able to present knowledge from a diversity of sources with clarity as to its derivation, consequences and structural relations
- Users of the knowledge acquisition system should be able to apply the knowledge in a variety of familiar domains and freely experiment with its implications
- The system should make provision for algorithmic expression of knowledge where appropriate
- The system should make provision for validation studies
- As much of the operation of the system as possible should be founded on well-developed and explicit theories of knowledge acquisition, elicitation and representation
- As the overall knowledge acquisition develops it should converge to an integrated system.

All of these requirements are subject to caveats--some domain dependence may be appropriate for efficiency in specific knowledge acquisition systems--some human intervention may be helpful or necessary when an expert is using the system--and so on. However, the broad design goals stated capture the key issues in the knowledge acquisition system design community.

A number of successful rapid prototyping systems widely used in industry have been based on the application of repertory grid techniques (52) to elicit an expert's conceptual framework. These systems have already gone through several generations from the original computerization of repertory grids (53) through the PLANET system, (54,55) the Expertise Transfer System, (56,57) KITTEN, (49) and AQUINAS.(50) The next section briefly describes KSSO, a fifth-generation development of such techniques as an integrated suite of knowledge acquisition programs.

KNOWLEDGE SUPPORT SYSTEM ZERO (KSSO)

Hays-Roth, Waterman & Lenat (58) have analyzed the way in which communication problems impede the process of transferring expertise into a program. They note that the vocabulary initially used by the expert to talk about the domain with a novice is often inadequate for problem solving, so that the knowledge engineer and expert must work together to extend and refine it. One of the most difficult aspects of the knowledge engineer's task is helping the expert to structure the domain knowledge, to identify and formalize the domain concepts. This elicitation of an adequate terminology, and its structuring, are the primary functions of prototyping tools based on repertory grid techniques.

The basic concepts underlying repertory grids are those already familiar in computing of entity-attribute databases. The expert is asked to define classes of entities that are relevant to different areas of the problem domain, and then to derive significant attributes of these entities by considering how distinctions are made between them. This process is best seen through some practical examples.

Figure 16 is the start-up screen for KSSO. The system is designed to be very clean and simple to use with a substantial entertainment value through the use of digital speech and animation. The objective is to make interaction with system attractive and fun while retaining a professional working environment. Spoken and visual help are available at all times, and context sensitive manual sections are stored on the bottom of each screen so that they are accessible through scrolling. The expert should never be at a loss, need help he cannot get, or have to access a manual.

The default action at all times is to click on the button at the top right of the lower control panel--usually "OK", "Yes", "More" or the most likely action. During elicitation, when the user has a choice of actions, this default action will select one by biased random choice from a reasonable set generated by rules that can be edited by the knowledge engineer. Thus, experts can begin to make effective use of the system very quickly with no need for computer experience and very little hand holding.

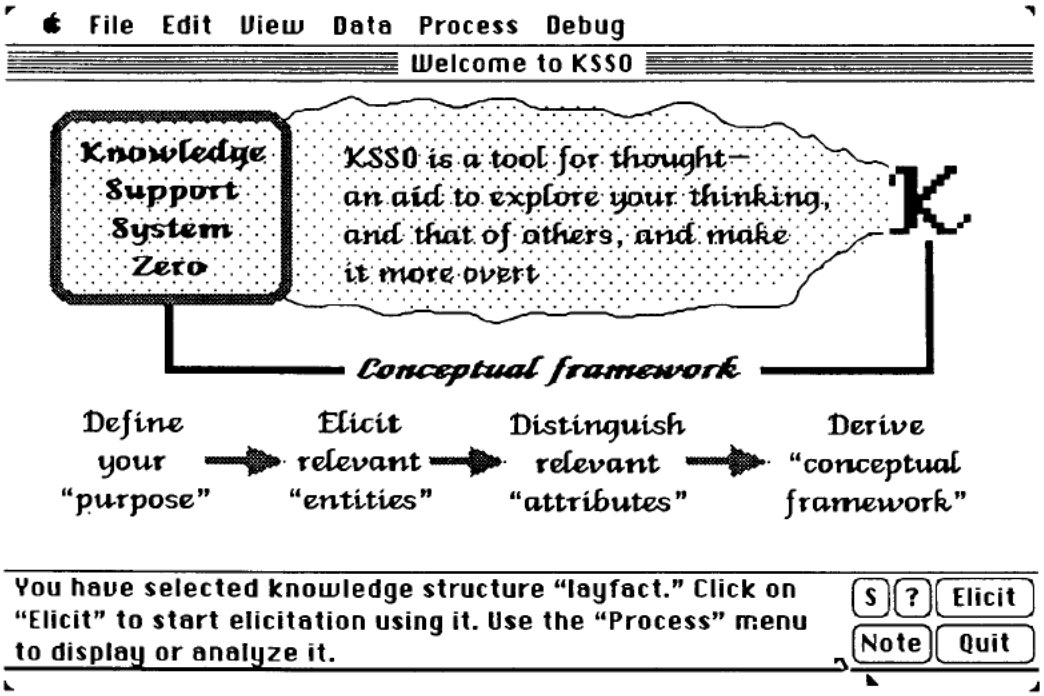

Figure 16. KSSO start-up screen showing process of elicitation.

Figure 17 is an entity screen from KSSO in a knowledge elicitation sequence concerned with factory layout planning.(59) KSSO allows its vocabulary to be tailored to the problem to maximize expert involvement, and the expert and knowledge engineer have agreed on a basic terminology. The expert has entered various "factors" which affect layout--these are the entities, and we will see the system help to elicit relevant "qualities"--these are the attributes.

The system keeps track of matches between entities and between attributes which indicate structures in the knowledge, and the expert has decided to investigate one of these. The indicator at the lower left of Figure 17 shows the value of the highest match between entities. The expert has clicked on it to highlight the matching entities. He can use the "-" and "+" buttons to step backward and forward through the matches.

He next clicks on the "Break" button to see whether an attribute can be input that will reduce the match--that is, a counterexample to the inferred structure. KSSO takes him to the screen shown in Figure 18, which makes explicit the question he has to answer and shows how the match has come about. The horizontal bars show the ratings of the entities on the attributes, with those of the upper entity ("location of services") being shown by the upward markers, and those of the lower entity ("area of department") being shown by the downward markers.

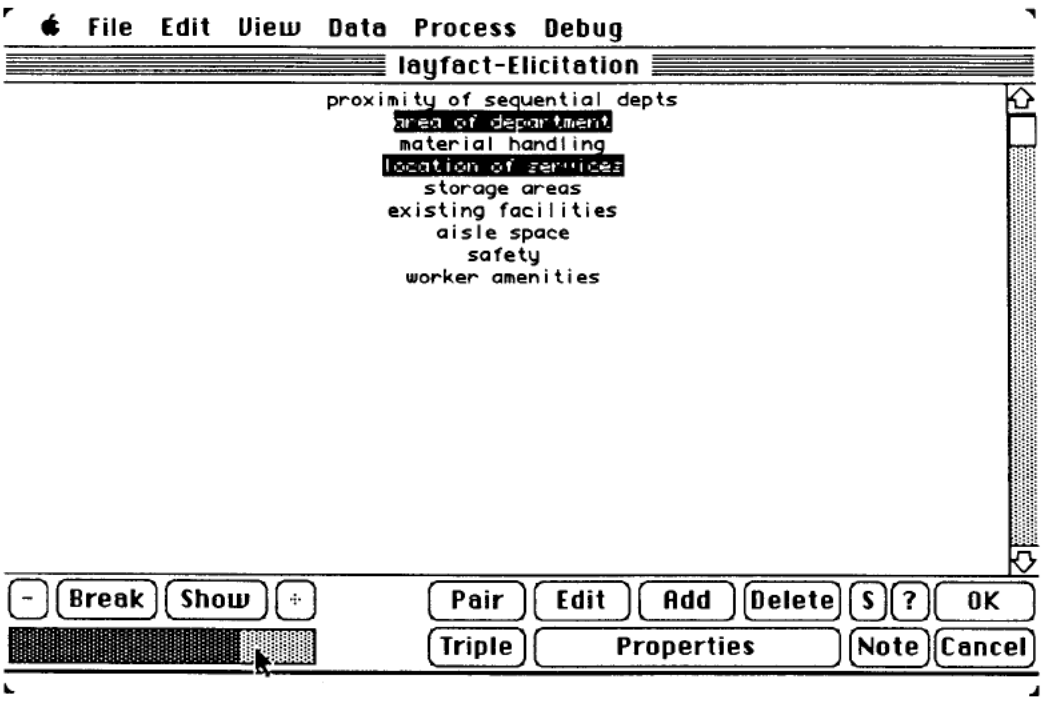

Figure 17. KSSO entity screen showing matching entities.

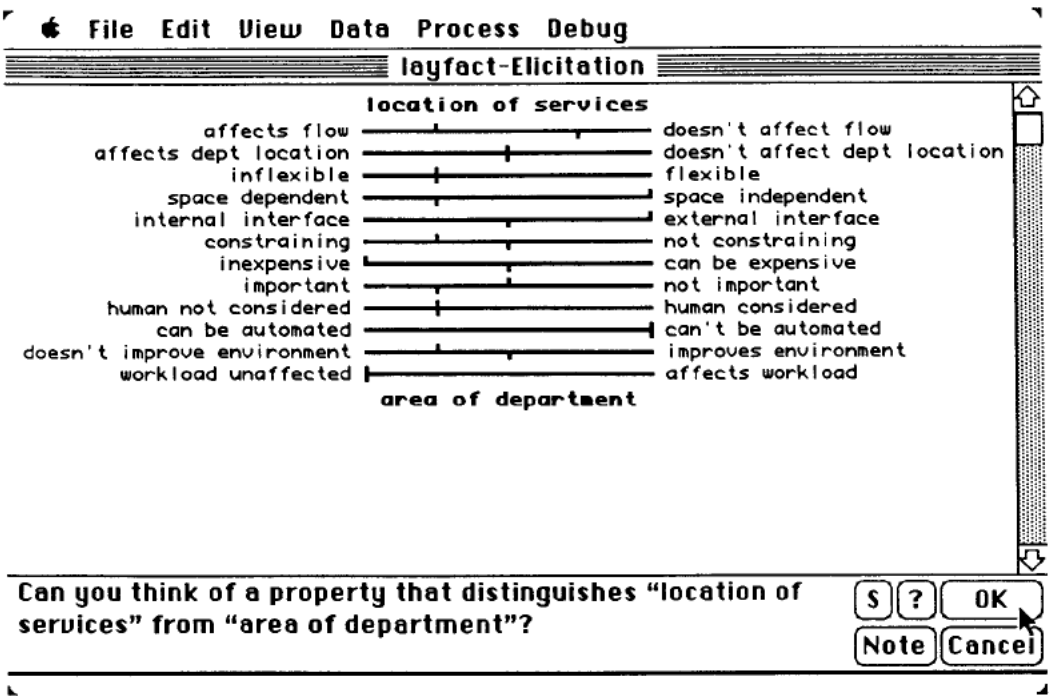

Figure 18. KSSO entity match screen prompting new attribute.

The expert can edit these ratings by dragging the markers along the scales, or modify the entity names by clicking on them to open up a text edit window. At no time in KSSO is he constrained to follow a specific course of action because the system is designed to facilitate free exploration and development of knowledge structures. In this case, the expert decides to complete the action by answering "OK" to the question asked. KSSO then takes him to the screen shown in Figure 19 where he is prompted to define a new attribute and then rate all the entities on it by clicking and dragging them to the vertical rating bar in the center.

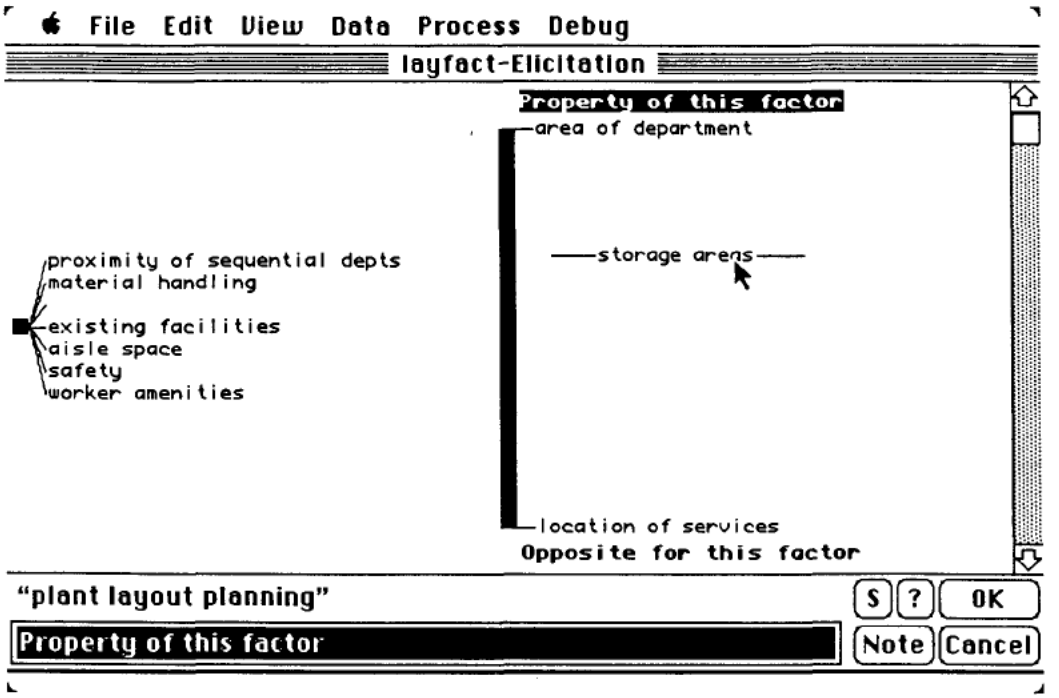

Figure 19. Screen on which new attribute is defined and all entities are rated.

There is a complementary sequence in which the expert starts from the attribute screen and clicks on the indicator to see the highest matched attributes. He then clicks on the "Break" button again and moves to a screen where the basis of the match is shown and again a very specific question is asked about the type of activity required to break the match. He can again edit the ratings by clicking and dragging the entities. Answering "OK" to this question takes him to a screen where the name of the new entity is entered and then rated on each of the existing attributes.

Figure 20 shows a different form of attribute elicitation initiated when the expert clicks on the "Triple" button in Figure 17.

KSSO generates a triple of three entities and asks him in what way two are alike and differ from the third. He clicks on "worker amenities" and KSSO takes him to the rating screen shown in Figure 21 with the triple pre-rated, so that he may enter the attribute and the ratings of the other entities.

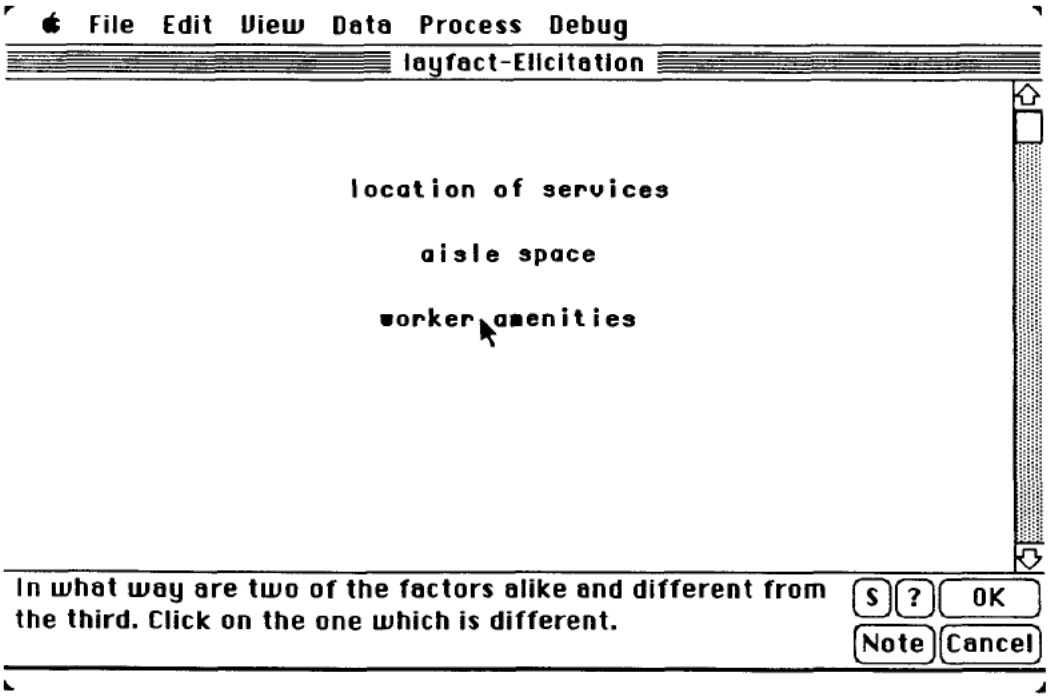

Figure 20. KSSO attribute elicitation from a triple of entities.

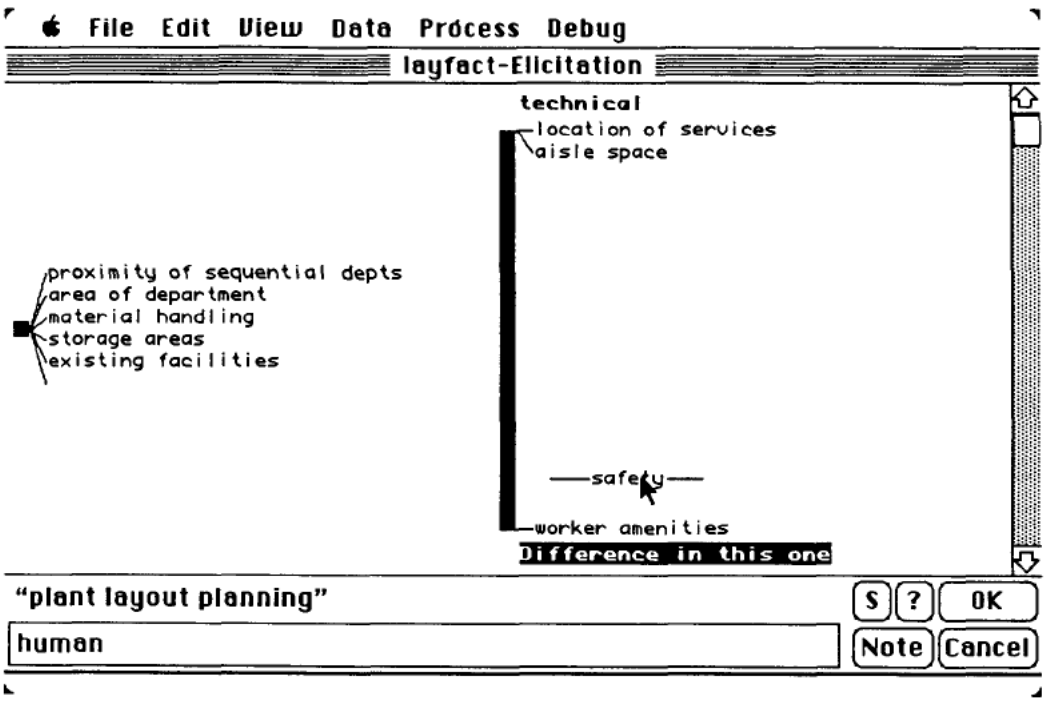

Figure 21. KSSO attribute entry and rating screen after triple.

The triples are generated on a random basis biased to promote variety, and this mode of elicitation tends to trigger recollections of significant attributes otherwise forgotten.

At any time during the elicitation the expert system can ask for the structure of his entity-attribute conceptual system to be analyzed. Figure 22 shows a hierarchical clustering of his entities and attributes based on the matches between them.

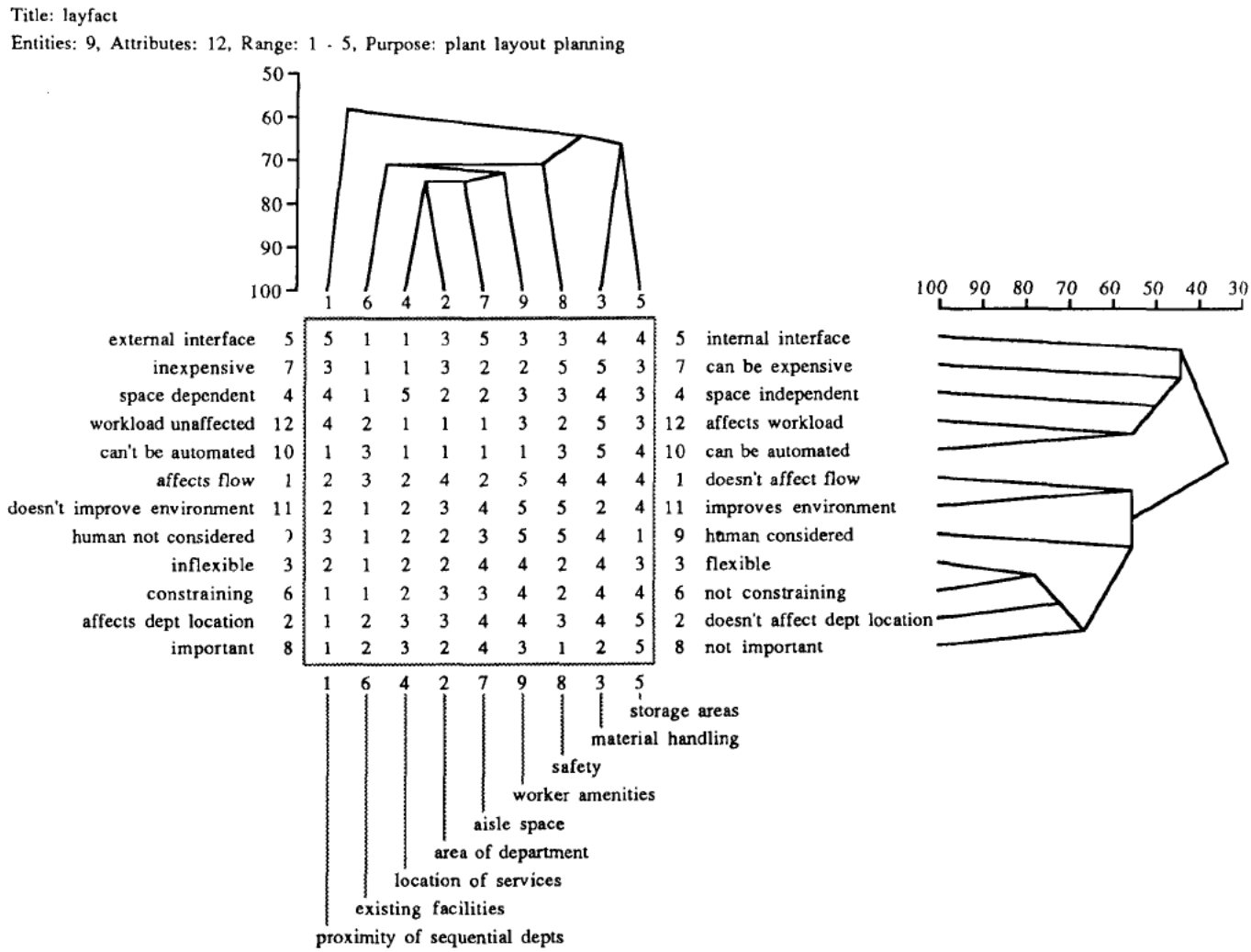

Figure 22. KSSO hierarchical clustering of attributes and entities.

Figure 23 shows an alternative mapping into two dimensions based on a principal components analysis.

Figure 24 shows a third form of analysis based the logical entailments in his data.(60,61) This third form is used to generate rules with which to prime an expert system shell.

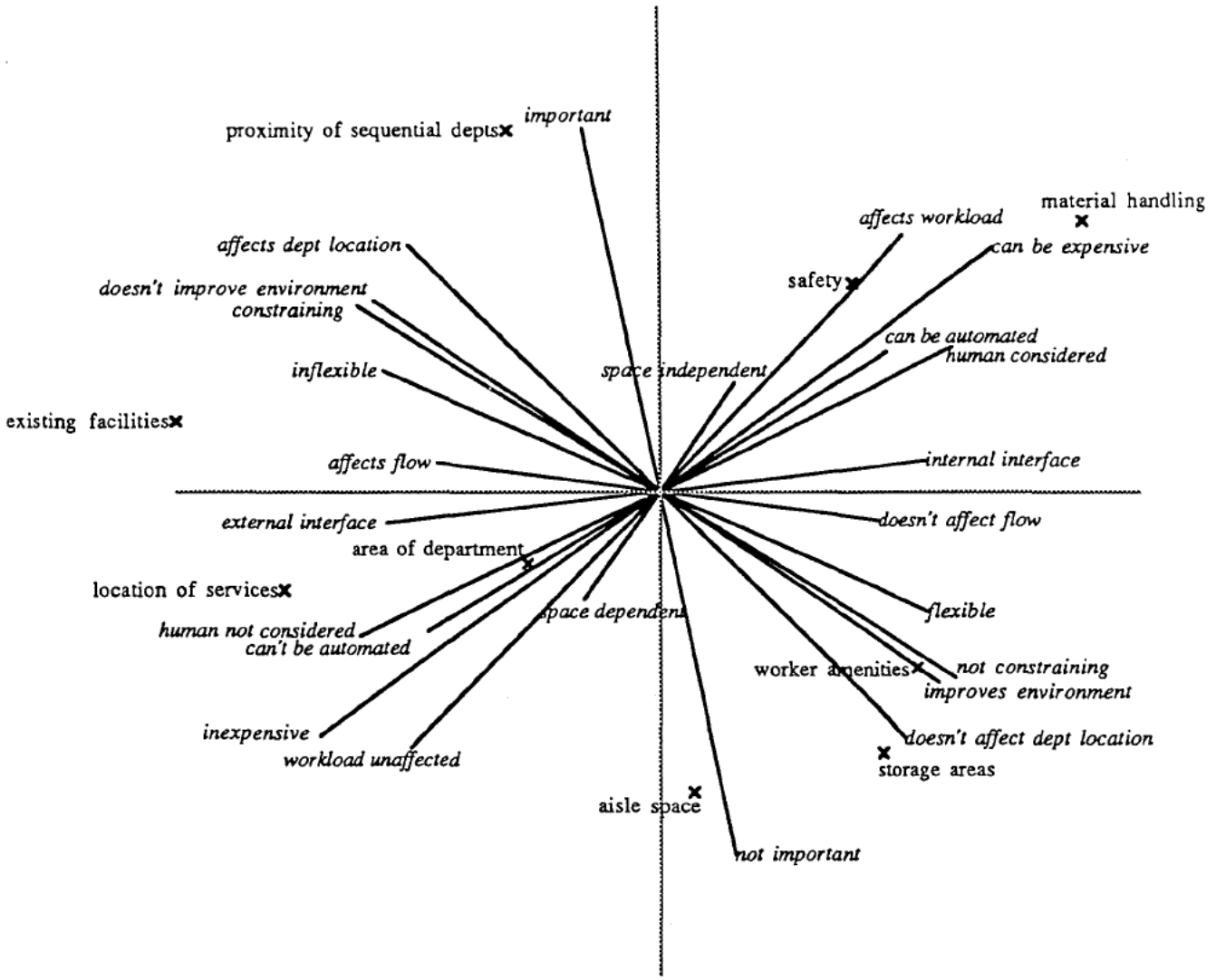

Figure 23. KSSO principal components map of attributes and entities.

Figure 25 shows the logical entailments being used within Nexpert in a prototype expert system based on the conceptual structures whose elicitation has been shown together with rules relating these concepts to layout decisions.

The technical details of the processes going on in Figures 16 through 25 are less important than the psychological impact on the expert. Users of KSSO generally find that it gives fascinating insights into their own thought processes, which both makes it attractive and puts them into an introspective frame of mind conducive to knowledge elicitation. Most of the dialog they see on the screen is generated from their own inputs and hence appears natural in terms of their professional terminology. Most importantly, this translates immediately into the rules generated for use in the expert system shell--they use the expert's jargon and their derivation from the data entered has already been shown to the expert and changes negotiated.

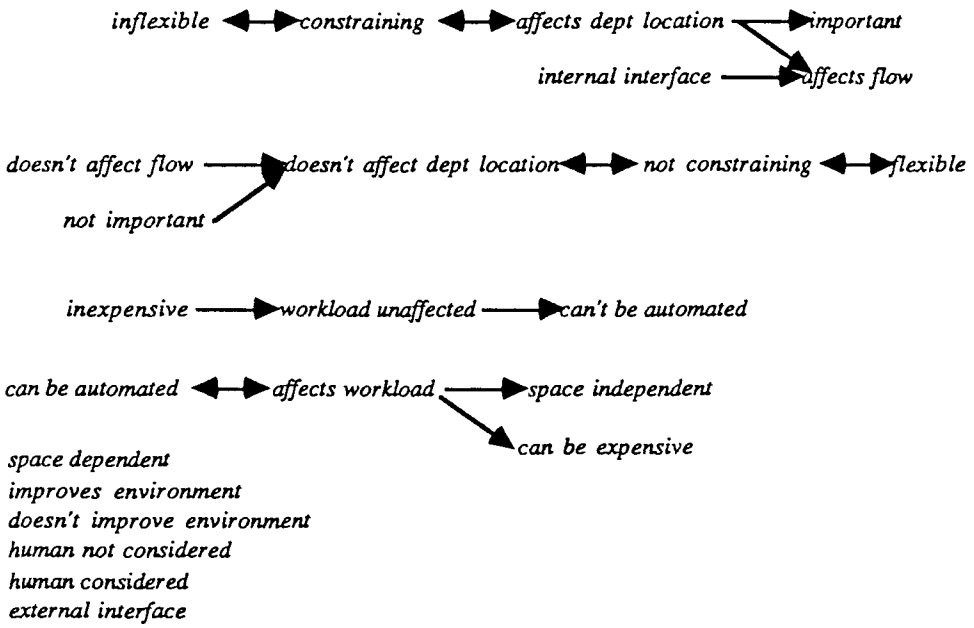

Figure 24. KSSO logical analysis of entailments between attributes.

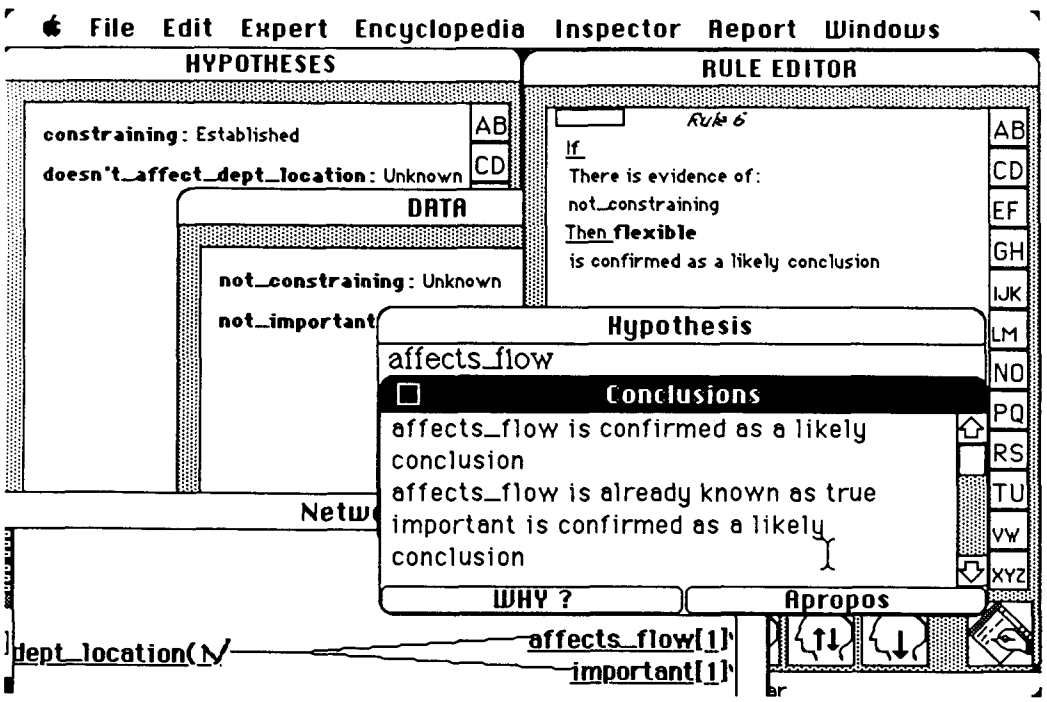

Figure 25. Nexpert operating with rules generated by KSSO.

Thus, the initial prototype system, such as that shown in Figure 25, is comprehensible to the expert in its derivation and terminology. Also, by that stage he has become accustomed to the use of the computer--windows, menus, and mice are familiar tools--and will often move over to editing the rules in the shell without really noticing that the environment has changed.

In most problem environments there are multiple knowledge islands, each requiring a separate elicitation process specific to the knowledge structures within the island. These tend to form a hierarchy, for example, "safety" is a general factor in the elicitation of Figure 17, but has its own detailed conceptual structure in terms of a variety of hazards, a variety of forms of accident, a variety of legal implications, etc. The methodology shown above can be extended to deal with such hierarchies.(62)

KSSO also contains facilities for extracting conceptual clusters by analyzing word associations in textual material. This is useful for defining knowledge islands, and pump-priming the related elicitations with entities and attributes. The logical entailment analysis in KSSO is a rule induction process that can be applied to examples of expert decisions encoded in entity-attribute form. Thus the three forms of expert system design shown in Figure 15 are incorporated in an integrated form in KSSO.

CONCLUSIONS

The foundations of manufacturing are changing as automation and flexibility restructure the nature of the manufacturing process and its management. The advantages of the new technologies bring with them severe problems of coping with a complex and rapidly changing technological environment. Knowledge-based systems, through their capabilities to integrate heterogeneous knowledge processes, offer the possibility of management decision support systems to aid the management of modern manufacturing.

The technologies underlying knowledge-based systems are well advanced, and it is possible to develop major decision support systems for manufacturing. The critical technologies to applications are the ESSE tools for interfacing with and validating the knowledge base, and the knowledge acquisition tools for transferring knowledge from human experts to the knowledge base.

ACKNOWLEDGMENTS

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

1. W.A. Sullivan, "Computer Integrated Manufacturing Program Justification," Proceedings of AUTOFACT '85 (Dearborn, MI: Society of Manufacturing Engineers, 1985), pp. 2-34 to 2-55.

2. M. Uenohara, "Flexible Automation in Japanese Electronics Industry," Proceedings of Japan-USA Symposium on Flexible Automation (Osaka, Japan: 1986), pp. 5-12.

3. T.G. Gunn, Computer Applications in Manufacturing (New York: Industrial Press, 1981).

4. C. Morgan, Robots: Planning and Implementation (Bedford, UK: IFS Publications, 1984).

5. A. Kusiak, A. Vannelli, and K.R. Kumar, "Grouping Problems in Scheduling Flexible Manufacturing Systems," Robotica, vol. 3, no. 4 (1985), pp. 245-252.

6. J.D. Elavia, "Where Do I Start with CIM?" Proceedings of AUTOFACT '85 (Dearborn, MI: Society of Manufacturing Engineers, 1985), pp. 1-1 to 1-21.

7. R.D. Miller, "Simplification, Management Control, Then Intelligent Automation," Proceedings of AUTOFACT '85 (Dearborn, MI: Society of Manufacturing Engineers, 1985), p. 1-14.

8. J.J. Nassr, "Considerations in Establishing a CIM Environment," Proceedings COMPINT'85, IEEE 85CH2136-0 (Montreal: 1985), pp. 68-72.

9. P.H. Francis, "Toward a Science of Manufacturing," Mechanical Engineering, vol. 108, no. 5 (May 1986), pp. 32-37.

10. G. Bruno, A. Elia, and P. Laface, "A Rule-based System for Production Scheduling," Proceedings COMPINT'85, IEEE 85CH2136-0 (Montreal: 1985), pp. 35-39.

11. S.H. Kim and N.P. Suh, "On an Expert System for Design and Manufacturing," Proceedings COMPINT'85, IEEE 85CH2136-0 (Montreal: 1985), pp. 89-95.

12. K.G. Kempf, "Manufacturing and Artificial Intelligence," Robotics, vol. 1, no. 1 (May 1985), pp. 13-25.

13. D. Liu, "Intelligent Manufacturing Planning Systems," SME Technical Paper MS85-1070, AUTOFACT '85 (Dearborn, MI: Society of Manufacturing Engineers, 1985), p. 1-17.

14. M.E. McDonald, "The Smart Foreman--Expert Systems for Manufacturing," SME Technical Paper MS85-1112, AUTOFACT '85 (Dearborn, MI: Society of Manufacturing Engineers, 1985), p. 1-9.

15. W.G. Sullivan and S.R. LeClair, "Justification of Flexible Manufacturing Systems Using Expert System Technology," Proceedings of AUTOFACT '85 (Dearborn, MI: Society of Manufacturing Engineers, 1985), pp. 7-1 to 7-13.

16. D. Ben-Arieh, "A Knowledge-based System for Simulation and Control of FMS," Simulation in Manufacturing, ed. R.D. Hurrion (Bedford, UK: IFS Conferences), pp. 287-295.

17. A. Kusiak, "Artificial Intelligence and Operations Research in Flexible Manufacturing Systems," Information Systems and Operational Research (INFOR), vol. 25, no. 1 (1987), pp. 2-12.

18. A. Kusiak, "Knowledge Engineering and Optimization in Automated Manufacturing Systems," Proceedings of the Second International Conference on Production Systems (Paris: INRIA, April 1987), pp. 15-28.

19. C.A. Smith, "The Role of Simulation Analysis in Improving the Manageability of Integrated Systems," Proceedings of AUTOFACT '85 (Dearborn, MI: Society of Manufacturing Engineers, 1985), pp. 11-1 to 11-10.

20. D.B. Wortman and R.J. Miner, "The Role of Simulation in Designing Manufacturing Systems," Proceedings of AUTOFACT '85 (Dearborn, MI: Society of Manufacturing Engineers, 1985), pp. 11-11 to 11-20.

21. J. Shivnan and J. Browne, "AI-based Simulation of Advanced Manufacturing Systems," Simulation in Manufacturing, ed. R.D. Hurrion (Bedford, UK: IFS Conferences, 1986), pp. 297-309.

22. K.E. Wichmann, "An Intelligent Simulation Environment for the Design and Operation of FMS," Proceedings of the Second International Conference on Simulation in Manufacturing, ed. J.E. Lenz (Bedford, UK: IFS Conferences, 1986), pp. 1-11.

23. O. Maimon, "FMS Real Time Operational Control," Proceedings of AUTOFACT '85 (Dearborn, MI: Society of Manufacturing Engineers, 1985), pp. 6-31 to 6-58.

24. J.S. Rhodes, "The Development of Integrated Control Systems for Flexible Manufacturing Systems," Proceedings of AUTOFACT '85 (Dearborn, MI: Society of Manufacturing Engineers, 1985), pp. 6-11 to 6-29.

25. G. Spur and K. Mertins, "Interactive Production Control in CAM," Proceedings COMPINT'85, IEEE 85CH2136-0, (Montreal: 1985), pp. 572-576.

26. T. Brown, S.M. Alexander, V. Jagannathan, and R. Kirchner, "Demonstration of an Expert System for Manufacturing Process Control," AI, Graphics and Simulation, ed. G. Birtwistle (La Jolla, CA: Society for Computer Simulation, 1985), pp. 110-113.

27. R.H. Kirschbrown and R.C. Dorf, "KARMA--a Knowledge-based Robot Manipulation System," Robotics, vol. 1, no. 1 (May 1985), pp. 3-12.

28. M.L. Brodie and J. Mylopoulos, eds., On Knowledge-Base Management Systems: Integrating Artificial Intelligence & Database Technologies (New York: Springer-Verlag, 1986).

29. R. Frost, Introduction to Knowledge Base Systems (New York: MacMillan, 1986).

30. B.R. Gaines, "Expert Systems and Simulation in the Design of an FMS Advisory System," Simulation in Manufacturing, ed. R.D. Hurrion (Bedford, UK: IFS Conferences, 1986), pp. 311-324.

31. B.R. Gaines and M.L.G. Shaw, "Knowledge Engineering for an FMS Advisory System," Proceedings of the Second International Conference on Simulation in Manufacturing: AMS'86, ed. J.E. Lenz (Bedford, UK: IFS Conferences, 1986), pp. 51-61.

32. F. Wenstop, "Deductive Verbal Models of Organizations," Fuzzy Reasoning and its Applications, eds. E.H. Mamdani and B.R. Gaines (London: Academic Press, 1981), pp. 149-167.

33. B.R. Gaines and M.L.G. Shaw, "Foundations of Expert Systems," Management Decision Support Systems Using Fuzzy Sets and Possibility Theory, eds. J. Kacprzyk and R.R. Yager (Rheinland: ISR Interdisciplinary Systems Research, Verlag TUV, 1985), pp. 3-24.

34. R.A. Carringer, "Integrating Manufacturing Systems Using the Product Definition Interface," SME Technical Paper MS85-1111 (Dearborn, MI: Society of Manufacturing Engineers, 1985), pp. 1-17.

35. Y.W. Su, "Modeling Integrated Manufacturing Data with SAM," IEEE Computer, vol. 19, no. 1 (January 1986), pp. 34-49.

36. S. Kawabe, A. Okano, and T. Yoshida, "Robot Task Planning System Based on Product Modeling," Proceedings COMPINT'85, IEEE 85CH2136-0 (Montreal: 1986), pp. 471-476.

37. E.H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," Fuzzy Reasoning and Its Applications, eds. E.H. Mamdani and B.R. Gaines (London: Academic Press, 1981), pp. 311-323.

38. K.E. Peray and J.J. Waddell, The Rotary Cement Kiln (New York: Chemical Publishing Co., 1972).

39. I.G. Umbers and P.J. King, "An Analysis of Human Decision-making in Cement Kiln Control and the Implications for Automation," Fuzzy Reasoning and its Applications, eds. E.H. Mamdani and B.R. Gaines (London: Academic Press, 1981), pp. 369-381.

40. M.S. Fox, "Knowledge Representation for Decision Support, Knowledge Representation for Decision Support Systems, eds. L.B. Methlie and R.H. Sprague (Amsterdam: North-Holland, 1985), pp. 3-26.

41. F. Hayes-Roth, "The Industrialization of Knowledge Engineering," Artificial Intelligence Applications for Business, ed. W. Reitman (Norwood, NJ: Ablex, 1984), pp. 159-177.

42. Rappaport, "Task Analysis in Intelligent Systems Design," WESTEX-86, IEEE 86CH2332-5 (Anaheim, CA: June 1986), pp. 61-67.

43. B. Hayes-Roth, "A Blackboard Architecture for Control," Artificial Intelligence, vol. 26, no. 3 (July 1985), pp. 251-321.

44. R.S. Michalski and P.H. Winston, "Variable Precision Logic," Artificial Intelligence, vol. 29, no. 2 (August 1985), pp. 121-146.

45. J. Pearl, "Fusion, Propagation and Structuring in Belief Networks." Artificial Intelligence, vol. 29, no. 3 (September 1986), pp. 241-288.

46. W.R. Swartout, "Knowledge Needed for Expert System Explanation," Future Computing Systems, vol. 1, no. 2 (1986), pp. 99-114.

47. Rappaport, "Multiple Problem Spaces in the Knowledge Design Process," Proceedings of AAAI Workshop on Knowledge Acquisition for Knowledge-Based Systems (Banff: 1986), pp. 37-0 to 37-15.

48. B.R. Gaines, "An Overview of Knowledge Acquisition and Transfer," Proceedings of AAAI Workshop on Knowledge Acquisition for Knowledge-Based Systems (Banff: 1986), pp. 14-0 to 14-18.

49. B.R. Gaines and M.L.G. Shaw, "Knowledge of Engineering Techniques," Proceedings of AUTOFACT '86 (Dearborn, MI: Society of Manufacturing Engineers, 1986), pp. 8-79 to 8-97.

50. J.H. Boose and J.M. Bradshaw, "Expertise Transfer and Complex Problems Using AQUINAS as a Knowledge Acquisition Workbench for Expert Systems," Proceedings of AAAI Workshop on Knowledge Acquisition for Knowledge-Based Systems (Banff: 1986), pp. 5-0 to 14-16.

51. R.S. Michalski and R.L. Chilausky, "Knowledge Acquisition by Encoding Expert Rules versus Computer Induction from Examples--A Case Study Involving Soybean Pathology," International Journal of Man-Machine Studies, vol. 12 (1980), pp. 63-87.

52. G.A. Kelly, The Psychology of Personal Constructs (New York: Norton, 1955).

53. M.L.G. Shaw, On Becoming a Personal Scientist (London: Academic Press, 1980).

54. M.L.G. Shaw and B.R. Gaines, "A Computer Aid to Knowledge Engineering," Proceedings of British Computer Society Conference on Expert Systems (Cambridge: 1983), pp. 263-271.

55. M.L.G. Shaw and B.R. Gaines, "Interactive Elicitation of Knowledge from Experts," Future Computing Systems, vol. 1, no. 2 (1986), pp. 151-190.

56. J.H. Boose, "Personal Construct Theory and the Transfer of Human Expertise," Proceedings AAAI-84 (American Association for Artificial Intelligence, 1984), pp. 27-33.

57. J.H. Boose, "A Knowledge Acquisition Program for Expert Systems Based on Personal Construct Psychology," International Journal of Man-Machine Studies, vol. 23, no. 5 (November 1985), pp. 495-525.

58. F. Hayes-Roth, D.A. Waterman, and D.B. Lenat, eds., Building Expert Systems (Reading, MA: Addison-Wesley, 1983).

59. J. Adler and J. Matheson, "Expert Knowledge Elicitation Using Personal Construct Theory: Obtaining Plant Layout Expertise with 'PLANET'," BASc Thesis (Toronto: University of Toronto, Department of Industrial Engineering, 1986).

60. B.R. Gaines and M.L.G. Shaw, "New Directions in the Analysis and Interactive Elicitation of Personal Construct Systems," Recent Advances in Personal Construct Technology, ed. M.L.G. Shaw (London: Academic Press, 1981), pp. 147-182.

61. B.R. Gaines and M.L.G. Shaw, "Induction of Inference Rules for Expert Systems," Fuzzy Sets and Systems, vol. 18, no. 3 (April 1986), pp. 315-328.

62. J.H. Boose, "Rapid Acquisition and Combination of Knowledge from Multiple Experts in the Same Domain," Future Computing Systems, vol. 1, no. 2 (1986), pp. 191-216.