AD-A223 035

BDM

A Ford Aerospace Company

BDM

A Ford Aerospace Company

BDM INTERNATIONAL, INC.
1300 NORTH 17TH STREET
SUITE 950
ARLINGTON, VIRGINIA 22209
(703) 247-0340

Contract No.: DAAL03-87-C-0069
Contract Start Date: 9/14/90
Contract End Date: 12/31/90

Principal Investigator:
Dr. Brian G. Kushner
(703)247-0350

BDM-KAT: FINAL TECHNICAL REPORT OF RESEARCH RESULTS

March 31, 1990

BDM/ROS-90-0562-TR

Prepared by:
Dr. Juliana S. Lancaster
Dr. Brian G. Kushner
BDM International, Inc.
1300 N. 17th Street, Suite 950
Arlington, VA 22209

Sponsored by:

DARPA
1400 Wilson Blvd.
Arlington, VA 22209

ARO
P.O. Box 12211
Research Triangle Park, NC 27709-2211

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) BDM/ROS-90-0562-TR	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION BDM International, Inc. 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Army Research Office
6c. ADDRESS (City, State, and ZIP Code) 1300 N. 17th Street Suite 950 Arlington, VA 22209	7b. ADDRESS (City, State, and ZIP Code) P.O. Box 12211 Research Triangle Park, NC 27709-2211
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Defense Advanced Research Projects Agency 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAL03-87-C-0019
8c. ADDRESS (City, State, and ZIP Code) 1400 Wilson Blvd. Arlington, VA 22209	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) BDM-KAT - Final Technical Report of Research Results	
12. PERSONAL AUTHOR(S) Drs. J. S. Lancaster and B. G. Kushner	
13a. TYPE OF REPORT Final 13b. TIME COVERED FROM 9/14/87 TO 12-31-89	14. DATE OF REPORT (Year, Month, Day) 1990, March 31 15. PAGE COUNT
16. SUPPLEMENTARY NOTATION	
17. COSATI CODES FIELD / GROUP / SUB-GROUP	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The BDM-KAT project was initiated to address specific, critical needs for rapid development of knowledge bases to support intelligent materials processing systems. An initial premise of the BDM-KAT project was that experts across the general field of materials processing use common types of knowledge and similar reasoning mechanisms. For this reason, we proposed the development of a tool specifically oriented toward knowledge engineering for materials processing. Two versions of BDM-KAT were developed from different underlying paradigms. The lessons learned from the first version were of significant valued in developing version 2.0. The 2.0 version of BDM-KAT was developed from a cognitive psychology model of human knowledge representation and reasoning based on the general body of literature in human cognition and a specific model developed by Dr. L.M. Vekker. Three tools, Categorizer, Time Mapper, and Causal Charter have been prototyped and used in preliminary knowledge acquisition for an intelligent process controller for Hot Isostatic Pressing (HIP). Both the volume of information collected and structured and the value of that knowledge for the developing controller attest to the value of the concepts implemented in BDM-KAT version 2.0.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. David Hislop	22b. TELEPHONE (Include Area Code) (919) 549-0641 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

When reproduc[illegible]

BDM–KAT: Final Technical Report of Research Results

CONTRACT NUMBER: DAAL03–87–C–0019
TECHNICAL CLIENT: Army Research Office/DARPA

EXECUTIVE SUMMARY

The BDM–KAT project was initiated to address specific, critical needs for rapid development of knowledge bases to support intelligent materials processing systems. These needs have developed from the demand for decreases in the time lag between laboratory development of new materials or processing capabilities and their use in operational production environments and demands for increases in the quality and yield of existing processes. An initial premise of the BDM–KAT project was that experts across the general field of materials processing use common types of knowledge and similar reasoning mechanisms. For this reason, we proposed the development of a tool specifically oriented toward knowledge engineering for materials processing. Because, at the time of project inception, no working systems for intelligent materials processing had been developed or fully designed, we were able to address the specific domain issues that impinge on appropriate knowledge acquisition efforts independently of defacto or default representation and inferencing conventions. Furthermore, the BDM–KAT project and a separately funded intelligent controller development project were initiated simultaneously to allow rapid feedback between the knowledge elicitation efforts and the application requirements, to the advantage of both projects. BDM–KAT gained significantly from this feedback process in that, as we gained experience with intelligent materials processing and the Hot Isostatic Processing domain, characteristics and functional expectations of the tool were modified to accomodate the specific characteristics and requirements of the IPM domain. Two versions of BDM–KAT were developed from different underlying paradigms. The lessons learned from the first version were of significant valued in developing version 2.0. For both versions, the overall goals were to:

- accelerate the knowledge elicitation process;
- support incremental knowledge elicitation;
- enable full representation of the overall process being considered;
- facilitate the interactions of multiple experts; and
- aid in subsequent updates of materials knowledge bases.

BDM–KAT version 1.0 was developed from a software engineering paradigm. In its Clarification Mode, experts were led to develop a process model specification in terms of logical AND/OR connections between process stages and objects. In the Prediction Mode, the process model was automatically reframed as a flow model and the expert was queried about specific predicted process states, acceptable and unacceptable sensor val-

ues as specific process points, and missing components of the process. In the Diagnosis Mode, the expert was to observe a step-through simulation of the process to identify additional missing components or improperly specified relationships between process components.

BDM-KAT 1.0 was successful in that the software model was demonstrated to be a useful mechanism for process specification. However, as a usable knowledge engineering tool, BDM-KAT 1.0 was flawed because:

- The screen representations were unfamiliar to our domain experts.
- The transitions between modes during sessions were excessively slow.
- Management of the growing library of process objects was difficult and inefficient.
- The tools were not portable, requiring the experts to travel to our facilities for all sessions.
- The tools were not capable of capturing heuristic dialogue between expert and knowledge engineer.

Given this experience and these lessons, a second version of BDM-KAT was developed from a different theoretical paradigm. For this version, these goals were added to the original set of program goals:

- reducing the requirements for translations of knowledge, particularly by the domain expert;
- reducing the burdens in time and effort on the domain expert;
- reducing the dependence on the skill of the individual knowledge engineer;
- providing a mechanism for selecting appropriate techniques; and
- providing an overall control structure for a knowledge elicitation effort.

The 2.0 version of BDM-KAT was developed from a cognitive psychology model of human knowledge representation and reasoning based on the general body of literature in human cognition and a specific model developed by Dr. L.M. Vekker. In this version of BDM-KAT, the individual tools and user interface are designed to maximize compatiblity with the domain expert's own internal representations, thus making the elicitation process more comfortable for the expert. In addition, the tools are implemented in the C programming language on SUN workstations to allow portability to the expert's own working location. Three tools, Categorizer, Time Mapper, and Causal Charter have been prototyped and used in preliminary knowledge acquisition for an intelligent process controller for Hot Isostatic Pressing (HIP). The tools were used in a total of twelve hours of active knowledge elicitation for the HIP application. Both the volume of information collected and structured and the value of that knowledge for the developing controller attest to the value of the concepts implemented in BDM-KAT version 2.0.

Based on the utility of the tools in actual knowledge elicitation and the comments of domain experts during and after their sessions with the tools, we believe that this research effort has succeeded. We have prototyped a set of tools that reflect an approach to knowledge acquisition that addresses many of the shortcomings of more traditional techniques. Specifically, we believe BDM-KAT 2.0 can:

improve the nature of the domain expert's experience with knowledge engineering;
streamline the overall knowledge acquisition process;
improve the structure of large-scale knowledge engineering efforts; and
enhance the verifiability of elicited knowledge.

EXECUTIVE SUMMARY

BDM–KAT: Final Technical Report of Research Results

CONTRACT NUMBER: DAAL03–87–C–0019
TECHNICAL CLIENT: Army Research Office/DARPA

I. PROGRAM GOALS AND MOTIVATION

The BDM–KAT project was initiated to address specific, critical needs for rapid development of knowledge bases to support intelligent materials processing systems. These needs have developed from the demand for decreases in the time lag between laboratory development of new materials or processing capabilities and their use in operational production environments and demands for increases in the quality and yield of existing processes. An initial premise of the BDM–KAT project was that experts across the general field of materials processing use common types of knowledge and similar reasoning mechanisms. For this reason, we proposed the development of a tool specifically oriented toward knowledge engineering for materials processing. Because, at the time of project inception, no working systems for intelligent materials processing had been developed or fully designed, we were able to address the specific domain issues that impinge on appropriate knowledge acquisition efforts independently of defacto or default representation and inferencing conventions. Furthermore, the BDM–KAT project and a separately funded intelligent controller development project were initiated simultaneously to allow rapid feedback between the knowledge elicitation efforts and the application requirements, to the advantage of both projects. BDM–KAT gained significantly from this feedback process in that, as we gained experience with intelligent materials processing and the Hot Isostatic Processing domain, characteristics and functional expectations of the tool were modified to accomodate the specific characteristics and requirements of the IPM domain. The following sections address these sets of requirements.

I.1 Intelligent Control Related Demands on Knowledge Engineering

Systems that control other systems and their processes have evolved through several technological stages. While a definitive taxonomy of control systems is not available and will not be attempted here, the broad range is characterized by: open loop control, state regulation, adaptive regulation and intelligent control. Many varieties of control systems, over a great range of complexities, can be classes as closed loop regulators. A system that accounts for conditions that are of higher order than the direct process state is considered adaptive. Intelligent control, such as that intended by the IPM program, implies more than adaptive control. The original concepts stemmed from expectations that intelligent control technology would yield tools allowing operators to actively control materials properties during production and generate decisions based on information derived from sensors and process models. Figure 1 shows the structure of an linear feedback control system on the left, and the added capabilities required for an intelligent control

system on the right. In practice, a large set of possibilities in architecture and technology

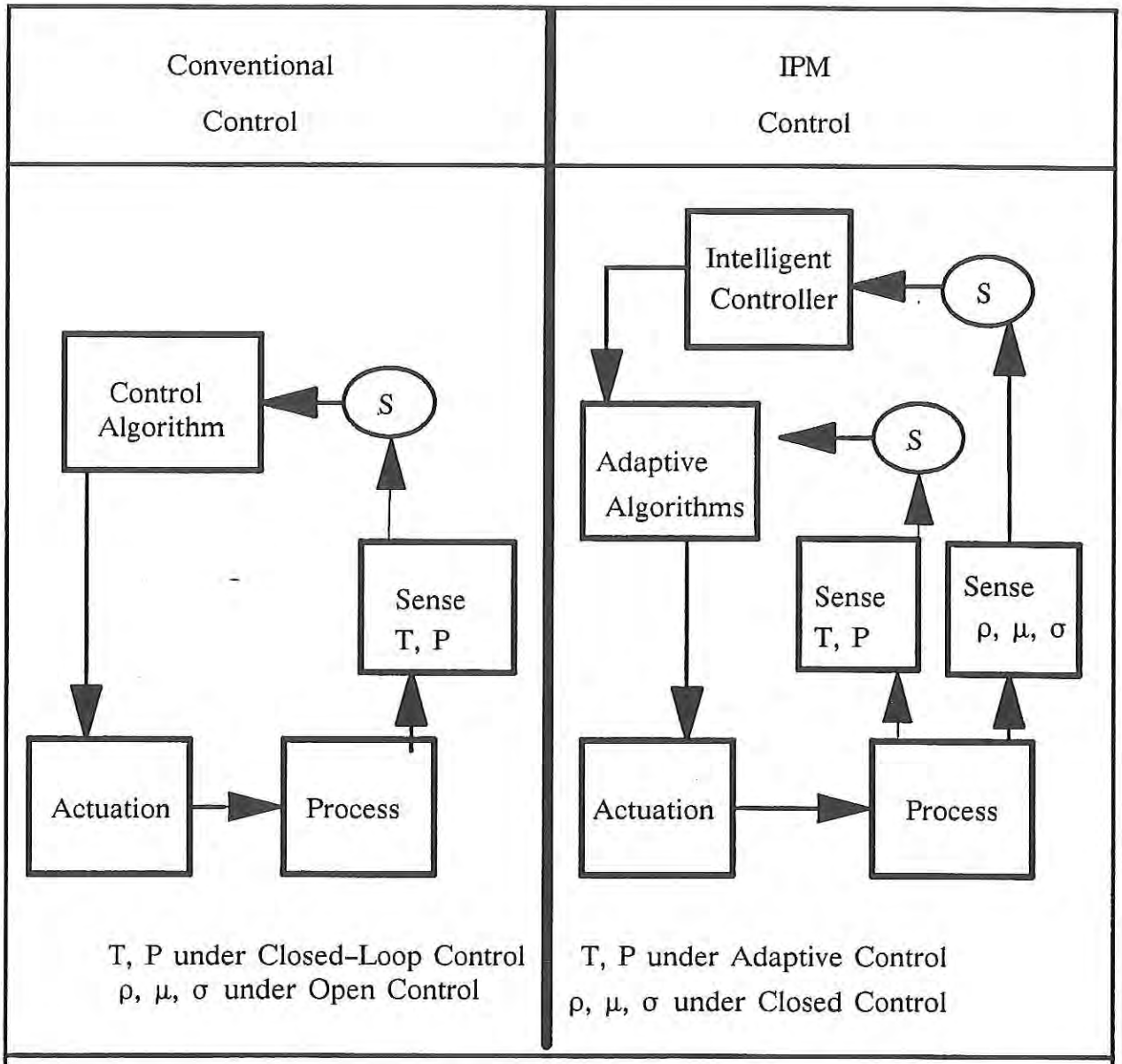


FIGURE 1 TRADITIONAL AND INTELLIGENT CONTROL SCHEMES

use is open for achieving the goals of IPM. Figure 2 shows a partial tree of the choices available to the builders of IPM systems and some of the knowledge acquisition techniques required to support them. BDM's choices from among this space are highlighted.

Using the functionality shown in Figure 1 and a *hybrid* (i.e., mixed knowledge-based and numeric processing) approach, BDM has developed and demonstrated a prototype intelligent controller for Hot Isostatic Pressing (HIP) of intermetallic powders under separate funding. The specific approach has been driven by the types of knowledge used by materials scientists in processing work and by close communication between available

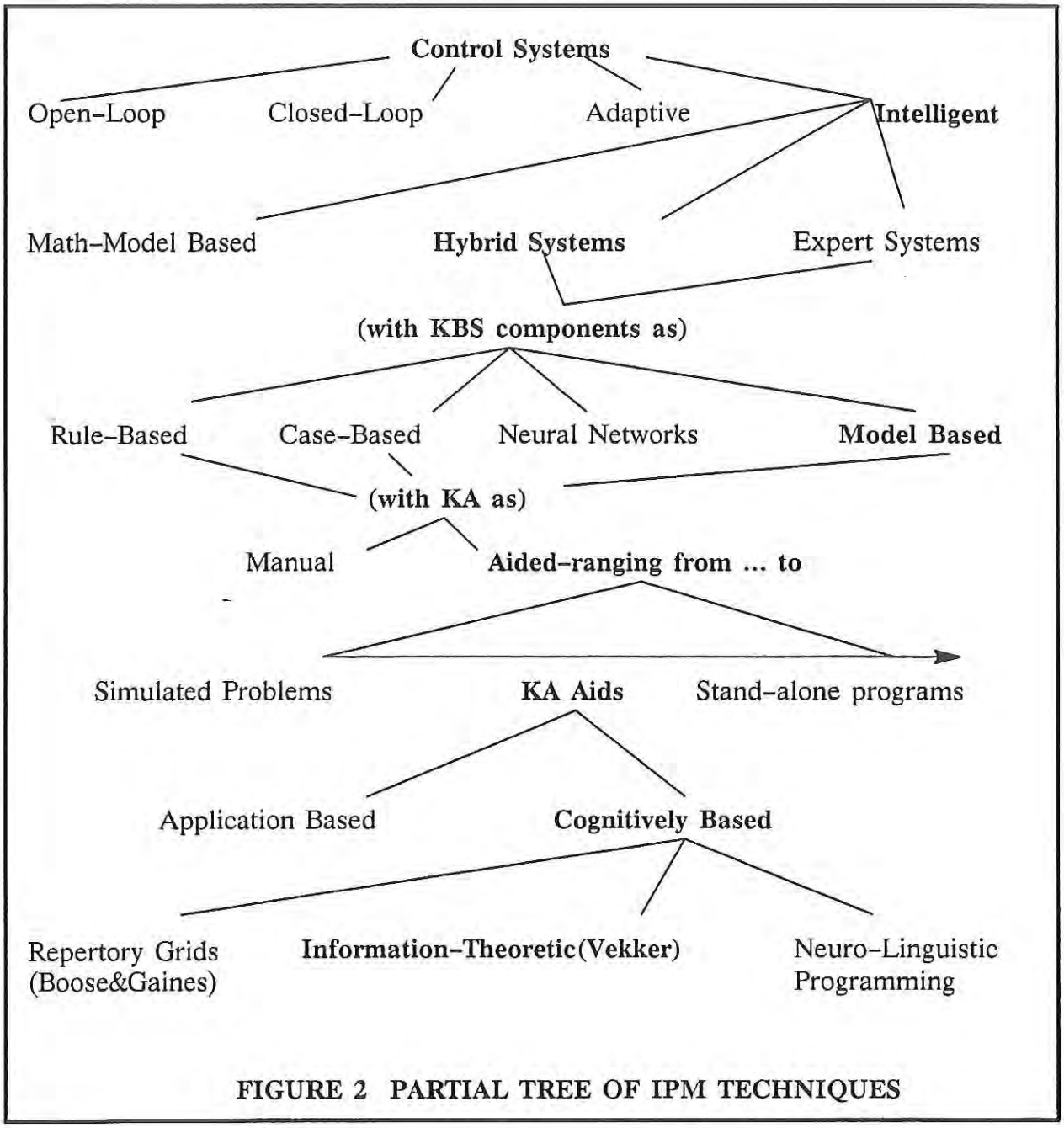


FIGURE 2 PARTIAL TREE OF IPM TECHNIQUES

quantitative models and this knowledge. The overall success of this controller, both in development time and performance capability serves as a demonstration that the knowledge acquisition techniques instantiated in BDM-KAT are successful and valid. The resulting system provides intelligent control of the HIP process, broadly defined, by supporting a set of activities that include:

- development of process schedules;
- prediction of process trajectories;

- observation of actual trajectories;
- comparison of actual and projected trajectories; and
- planning and executing corrective actions.

This design supplies future users with a full set of capabilities that may be needed by either a product/process designer or a real-time operator with control demands.

Within this framework, the knowledge elicited from domain experts and stored within the knowledge base can be used at a variety of points in the process to meet the changing goals of the ongoing process as shown in Figure 3. Initially, the knowledge, and

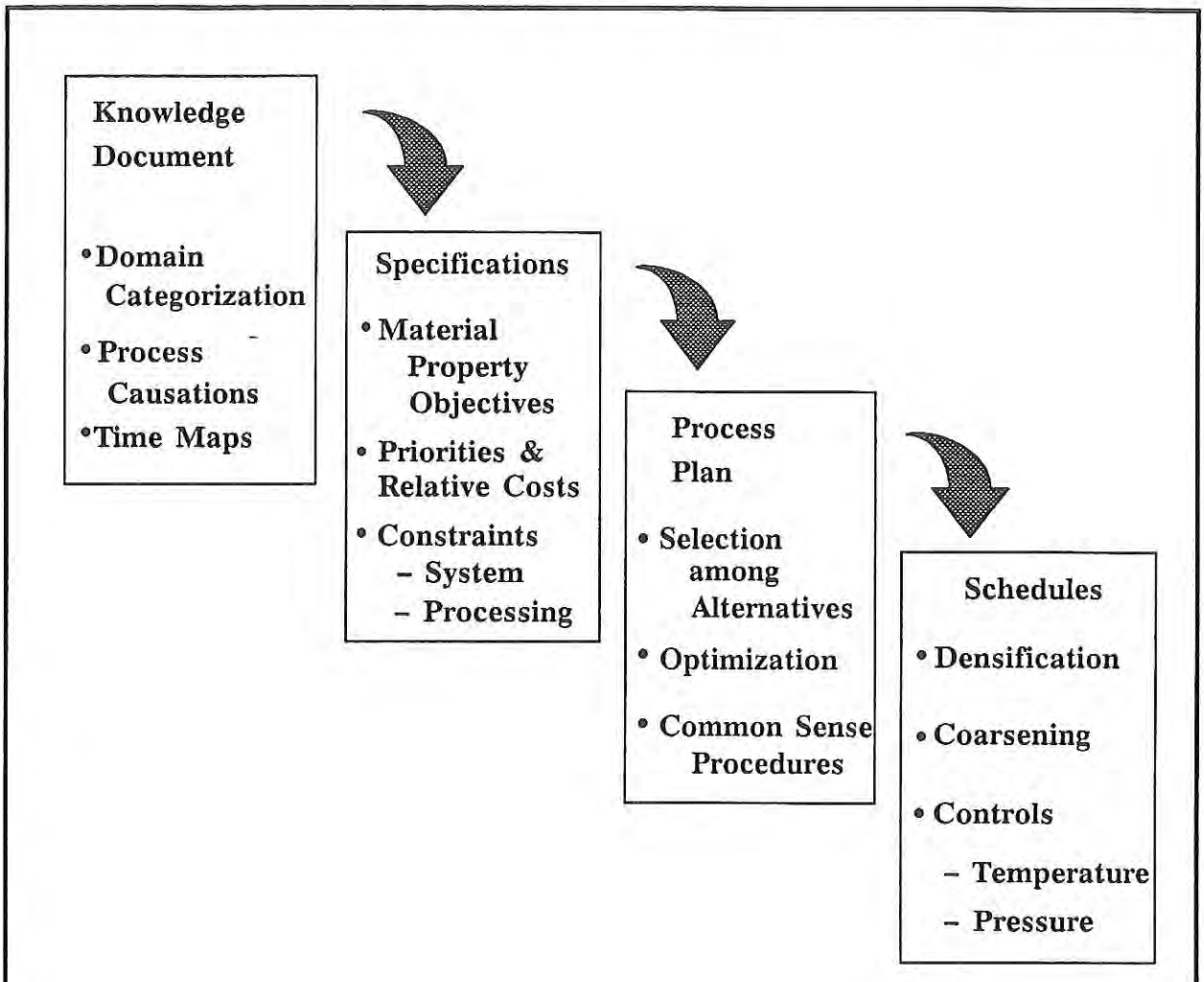


FIGURE 3 USES OF KNOWELDGE IN IPM

the elicitation process itself, are used to capture and represent the categorical, causal, and temporal structures implicit in the expert's knowledge as well as the general content knowledge. With these structures captured and integrated, the knowledge base can be used by a process or part designer to specify fully the objectives, priorities, and constraints such as those shown in Figure 4 that must be considered and/or met during process execution. During process planning, the specifications generated using the plan-

- **Material Property Objectives**
  - -- Final density ρ_{final}
  - -- Final grain size d_{final}

- **Priorities (Relative Costs)**
  - -- time w_{time}
  - -- diffusion depth w_{dd}
  - -- grain growth w_{gg}
  - -- energy consumption w_{energy}

- **HIP System Constraints**
  - -- Maximum temperature T_{Max}
  - -- Maximum pressure P_{Max}
  - -- Maximum ramp rates $\dot{T}_{Max}$, $\dot{P}_{Max}$

- **Processing Constraints**
  - -- Avoid densification front $\dot{\rho}_{Max}$, ∇T_{Max}
  - -- Avoid canister buckling ∇P_{Max}
  - -- Avoid excessive shape distortion $\dot{\rho}_{Max}$, ∇T_{Max}

FIGURE 4 HIP PROCESS SPECIFICATION

ner's own knowledge and the elicited knowledge base are used to guide generation of an optimum process schedule by a well-tuned set of quantitative models. During execution, the same knowledge is used to aid the operator in detecting and diagnosing possible property-related deviations in the behavior of the material or the equipment during processing and to suggest appropriate corrective actions as necessary.

I.2 Critical Characteristics of Materials Processing

There are unique challenges in the development of knowledge documents, domain models and intelligent control systems for materials processing domains such as HIP because the knowledge held by the various domain experts is of numerous types (e.g., quantitative models, phase diagrams, metallography, heuristics) and is changing rapidly

over time as new aspects of the domain are studied and understood. Consequently, a knowledge acquisition effort in IPM must confront the challenges of:

- the knowledge being neither stable nor well-formed.
- few "true" experts in the domain (with the few having limited time available to spend in knowledge acquisition).
- the lack of a single expert whose knowledge spans the full scope of the domain.
- the intelligent control systems being designed to take advantage of not-yet-developed technical capabilities (e.g., new sensors)

Our approach to meeting these challenges has been two-phases. We first sought to identify the types of knowledge needed to support the knowledge base functions discussed in the previous sections. Given an understanding of the types of knowledge needed, we worked to develop a methodology and environment for active knowledge elicitation that will enable us to surmount the challenges common to all knowledge acquisition efforts and unique to IPM domains.

Our identification of the knowledge types required for IPM was based on preliminary knowledge acquisition activities and a review of the knowledge and reasoning used by materials scientists and process planners during their activities and on specification of the functions to be served by the knowledge base within the HIP controller framework. From this analysis, we determined that a knowledge engineering tool for IPM must be capable of capturing the following types of knowledge:

- Domain structure, or knowledge about materials properties and interrelationships and about novel materials, to support completion of product and process specification and diagnosis.
- Causal networks, or knowledge about process control and material behavior, to support selection of materials and active control and diagnosis.
- Constraints, or knowledge about the temporal aspects of the process and the validity or invalidity of process model components, to support schedule optimization and selection.

With these requirements identified, we explored options for the specific design and use of a set of tools for use by knowledge engineers. In the following section, we present some details of the specific design used for *BDM-KAT* and the underlying reasons for the design.

I.3 Status of Existing Knowledge Engineering Methods and Tools

Knowledge engineering as a field has developed from the demand created by the development of expert or knowledge-based systems. In such applications, it is necessary that the knowledge base used by the system be well-organized, consistent, and reasonably complete. This, in turn, requires that the process by which that knowledge is entered into

the knowledge base be equally well-organized and controlled. However, early efforts in knowledge acquisition were modeled on small scale projects in which a single domain expert and a single developer could interact to build the final project, allowing a certain informality and ad hoc set of procedures to exist. In larger scale projects, the ad hoc nature of early knowledge engineering efforts was soon discovered to be limiting and unworkable.

In response to this situation, Hayes-Roth, et.al. (1983) presented a methodology for knowledge engineering. In this approach, a five-step cycle of identification, conceptualization, formalization, implementation, and testing is used iteratively until a satisfactory product results. While this provides a structure within which a system architecture and basic domain structure can be determined and allows for iteration through an identifiable sequence, it does not impose any control or guidance on the techniques used or support verification of the elicited knowledge. More current approaches tend to depict the knowledge engineering process as a more controlled, cyclic sequence, drawing on models and standards from the software engineering field. Some methodologies, like the adapted waterfall model of knowledge acquisition presented by McGraw and Harbison-Briggs (1989), suggest not only steps within the cycle, but also techniques for each phase and some guidance at decision and iteration points. However, even this structure does little to address how the knowledge acquisition task might be tied directly to the perceived structures and representations within the domain, as viewed by an expert.

In any manual knowledge engineering effort, the time and labor demands are high even when structures and practices are in place to direct and manage the timing and foci of knowledge engineering sessions. In addition, there are several pervasive and troublesome characteristics of the overall process. Specifically, true experts, because of the demand for their expertise are frequently unavailable for a series of several knowledge acquisition sessions, thus generating a demand on the knowledge engineer to maximize the value of every session held. When sessions are held, experts frequently have great difficulty accessing and then verbalizing significant components of their knowledge because it has been "chunked" or "proceduralized" during the development of expertise. Such knowledge is easily acted upon, but not easily discussed, making it difficult or impossible for the knowledge engineer to tap and depict the structure and relationships the expert holds of the domain. In addition, because such knowledge is, in many ways, implicitly assumed by the expert, it allows him/her to assume a level of understanding on the part of the knowledge engineer that is frequently inappropriate. Furthermore, many knowledge engineers cling to the interview as a primary technique. In the hands of a skilled communicator with adequate analytical skills and a good foundation in the domain, the interview can be highly useful. However, few knowledge engineers are able to use it efficiently, leading to large expenditures of time that result in little critical knowledge being elicited. This can be compounded by the practice of establishing the design and architecture of the target system in advance of knowledge acquisition--a practice that

forces a way of thinking about the domain and subdomains that may be incompatible with the model held by the domain expert. Finally, regardless of the techniques used or time expended, many manual-based knowledge acquisition efforts fail to document the knowledge elicited sufficiently for later verification and validation of the system.

A number of approaches to automating the process so as to alleviate some of these have been tried, ranging from simple problem simulations for discussion to fully automated knowledge acquisition environments. Most of these knowledge engineering tools have been developed to serve the needs of a specific system or a particular class of applications. In other words, they can be considered to be *implementation driven*. Because of their orientation toward the final representation and use of the knowledge by the system to be developed, the tools constrain domain experts by requiring that they present knowledge in the format on which the tool is based (e.g., repertory grids, fuzzy sets) or in the representation of the knowledge base (e.g., rules). This requirement may force the experts to transform the expression of their knowledge from a familiar, useful form to a novel, untested one. In contrast, many manual approaches do allow the expert to express knowledge in a more comfortable manner but place the burden for the transformation on the knowledge engineer.

I.4 Goals of the BDM-KAT Project

The original goals of this project were multi-faceted. As proposed, BDM-KAT was envisioned as an expert system for knowledge acquisition about materials processing that would necessarily include substantial general purpose knowledge about materials processing and use that knowledge to direct specific elicitation efforts. The value of such a system was described in terms of:

- accelerating the knowledge elicitation process;
- supporting incremental knowledge elicitation;
- enabling full representation of the overall process being considered;
- facilitating the interactions of multiple experts; and
- aiding in subsequent updates of materials knowledge bases.

During the research and development efforts of this project, a major shift in the underlying paradigm for BDM-KAT took place. While this change will be discussed in detail in later sections, it should be noted that none of the above goals were discarded. However, in redesigning BDM-KAT based on our growing experience with the domain, we focused our consideration on identifying a more global set of desired characteristics for a general purpose knowledge engineering environment and sought to develop an approach in line with them. Briefly, we determined that a knowledge engineering system should meet our earlier goals and provide significant support to the practicing knowledge engineer by:

- reducing the requirements for translations of knowledge, particularly by the domain expert;

- reducing the burdens in time and effort on the domain expert;
- reducing the dependence on the skill of the individual knowledge engineer;
- providing a mechanism for selecting appropriate techniques; and
- providing an overall control structure for a knowledge elicitation effort.

II. BDM-KAT: VERSION 1.0

II.1 Methodological Foundation: Software Engineering

The primary concept underlying the design of BDM-KAT 1.0 is that knowledge engineering and software engineering have much in common that can be taken advantage of to the improvement of both fields. The modes and interaction capabilities of BDM-KAT 1.0 were selected and designed to reflect some of the principles and methodologies of software engineering that have enhanced the specification and timely development of software process specifications.

The central thesis of this framework is that a computer program is. in and of itself, the specification of a process rather than as the embodiment of a problem solving methdology. Similarly, knowledge engineering for process control is fundamentally and exercise intended to produce a complete and logically coherent specification of a process and the reasoning supporting that process. Given these definitions of software and knowledge engineering, it becomes plausible to consider the ways in which the tools and techniques of software engineers could be used by knowledge engineers to ease their tasks. For additional detail concerning the underlying rationale and methodologies for BDM-KAT 1.0, see the attached papers by Blaxton & Reeker (1988); Kushner, Geesey, Parrish, & Wox (1987); Reeker, Blaxton, & Westphal (1988); and Westphal (1988).

II.2 Design and Implementation

BDM-KAT 1.0 was designed to support the knowledge engineer in building a knowledge base to be handed over to an expert system. The developed knowledge base would not include interface features or inference capabilities; these would require separate development. Building from the software engineering model, BDM-KAT 1.0 consisted of three query modes, each prompting the expert to answer different types o;f questions about the materials processing domain under elicitation and each using a different conceptualization and graphic display format of the process. The three modes were Clarification Mode, Prediction Mode, and Diagnosis Mode. Clarification and Prediction Modes were implemented using KEE and SIMKIT on a Symbolics 3670; Diagnosis Mode was designed, but only partial implementation was completed.

Clarification Mode provided the expert with facilities for creating and AND/OR graph representation of the process using a library of previously defined object types and rule classes. Prediction Mode redisplayed the content of the AND/OR model in a flow-

chart format and prompted the user to answer a series of questions about each subprocess represented in the structure. These queries would clarify acceptable ranges for sensor values and specific interactions between objects and would lead the expert to make predictions about possible failure points in the process and possible corrective or preventive actions at those points. Diagnosis Mode would be invoked once several Clarification–Prediction mode iterations had yielded a fairly detailed mode. In Diagnosis Mode, a step–through presentation of the modeled process would be executed, displaying all sensor readings defined. This mode allows the user to trace the pattern of rules being evoked and to diagnose the validity of the elicited model. In the following sections, each mode is described in some detail.

II.2.A The Object Library

Before a model could be built in BDM–KAT 1.0, a library was required that contained a sampling of the objects and rule classes expected to be required by the final model. The selection of components for this initial library was based on domain familiarization by the knowledge engineer through review of technical materials and preliminary conversations with domain experts.

II.2.B Clarification Mode

Clarification Mode may be thought of as a framework that allows the expert to tell the system what each process consists of, in terms of immediate sub–orocesses, as the expert sees them. This is a top–down, or process decomposition approach to modeling the knowledge. The result is an AND/OR graph, such as is shown in Figure 5. At the outset of Clarification Mode, certain objects corresponding to to units in the object library would be available in an object window. The expert may either use these objects or create now ones to be placed in the window and library. The expert moves a particular object into the main viewport and adds it to the developing AND/OR graph using a series of mouse actions. Once several objects have been placed in the main viewport, the expert can specify the relationships that hold between them, selecting from the set of relationships provided in the relations window. This process continues until the expert states that the process has been fully specified.

II.2.C Prediction Mode

Prediction Mode provides another perspective on the developing process model, transforming the AND/OR graph developed in Clarification Mode into a flowchart model. In this transformation, potential "holes" in the process model will be exposed which the expert can fill by either returning to Clarification Mode or adding objects directly to the Prediction Mode flowchart. While using Prediction Mode, the expert is queried about what sensors are used at each point in the process and what range of values is acceptable or indicative of trouble for each sensor. For such ranges of sensor values, the expert is guided through creation of a rule for the knowledge base using a Rule Builder tool that is

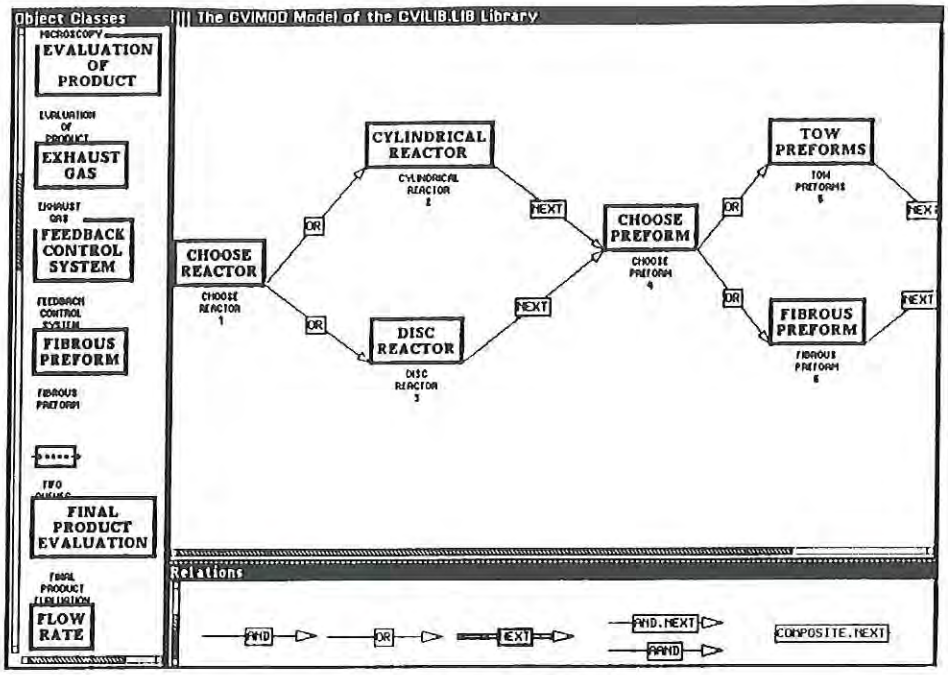


FIGURE 5

PARTIAL HIP PROCESS AS DISPLAYED IN CLARIFICATION MODE

executed within Prediction Mode. Figure 6 shows a sample Prediction Mode screen with a partially defined process.

II.2.D Diagnosis Mode

Once the expert is satisfied with the developed process model, most likely after several iterations through Clarification and Prediction Modes, Diagnosis Mode provides a means of "stepping through" the process in a dynamic manner, indicating consequences of the process in as much detail as is supported by the model. This step through allows the expert and knowledge engineer to identify additional, unsuspected gaps and/or errors in the model and to further specify rules and sensor requirements.

II.3 Results

The design and development of BDM–KAT 1.0 was the primary focus of the project through March of 1988. During that time, Clarification and Prediction Modes were fully implemented on a Symbolics 3670 and were transitioned into "production" versions through reimplementation on a different Symbolics machine. The initial implementation was developed using an unstructured rapid prototyping methodology; the "production" implementation was made using more structured methods and with an added focus on efficiency, documentation, and clarity of coding. Implementation proceeded fairly well, with the KEE and SIMKIT tools being highly useful within the Symbolics environment. The Diagnosis Mode was only partially implemented.

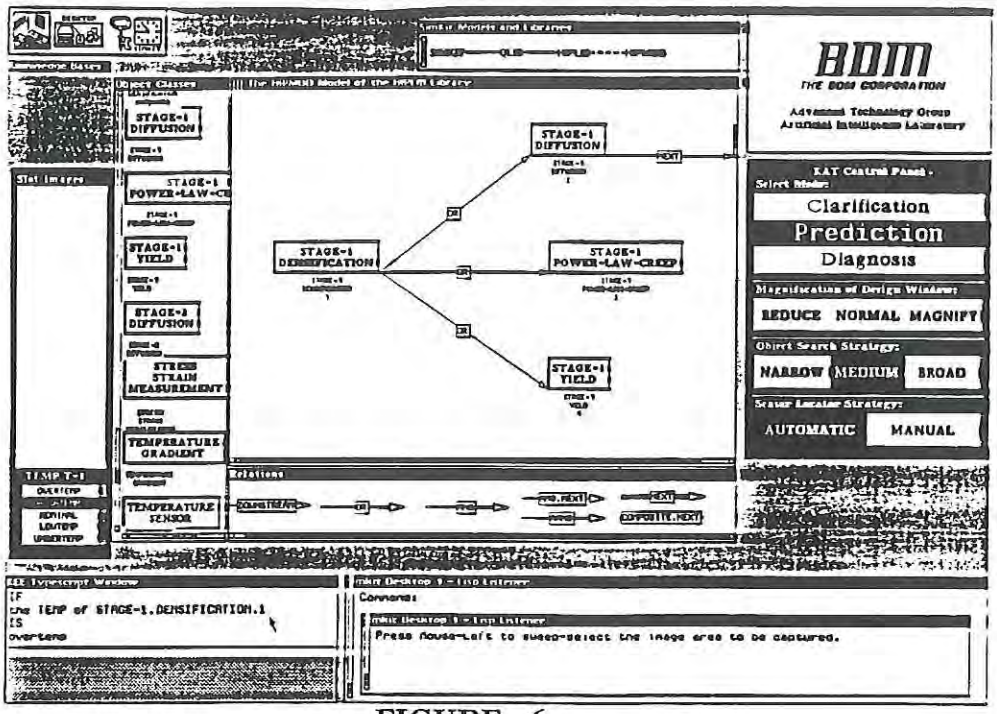


FIGURE 6
PARTIAL HIP PROCESS AS DISPLAYED BY PREDICTION MODE

Much of BDM–KAT 1.0 was highly successful. The automated rule builder implemented within Prediction Mode was highly useful for both experts and knowledge engineers. Limited implementation activities for Diagnosis Mode indicated that a step–through or simulation approach to knowledge verification would be extremely useful for process knowledge models, provided the simulation could be made to exercise the full model. From the perspective of the knowledge engineers and system developers, the AND/OR and flowchart representations were successful in capturing most aspects of the process models developed. The only area in which the representations were technically flawed was that not enough attention was paid to the hierarchical nature of the relationships between some objects. Consequently, the potential for confusion between levels of detail was present. This was not considered to be a serious problem, but rather to be something to be addressed as new object libraries were developed for new domains and processes. However, there were several severe flaws in BDM–KAT 1.0.

First, and most significant, our domain experts found the interface representations of Clarification and Prediction Modes to be unsatisfactory. A common comment was that "we don't think about it this way", and most experts refused to complete more that two sessions using BDM–KAT 1.0. This was a critical flaw, given our concerns that the knowledge be verifiable. The time involved for processing additions to the model and for transitioning between modes was excessive, especially given our avowed purpose of reducing the time requirements for knowledge engineering. As the size of the object library

grew, severe library management problems developed. Because the system was hosted on a Symbolics machine, it was not possible to install it on equipment in the workplace of our experts. This required that they come to our facilities for all knowledge engineering sessions, placing a severe time burden on them and exacerbating their overall disatisfaction with the tools. Finally, except within the limited boundaries of the Rule Builder, the tool was not capable of capturing any heuristic control knowledge expressed by the expert.

II.4 Lessons Learned

From this experience with BDM-KAT 1.0, we gained several significant insights. The comments of domain experts and increasingly apparent need to attend to hierarchical relationships in the process models led us to a better understanding of the types of knowledge that a system such as BDM-KAT would be required to elicit and represent. More specifically, we learned that the software engineering model of knowledge engineering was useful but not sufficient and did not lead to good interface representations from the expert's perspective. In addition, the comments of the experts underscored the importance of developing a system whose interface used a representational scheme compatible with he expert's own internal representations--both for gaining expert cooperation and for ease of knowledge elicitation.

III BDM-KAT: VERSION 2.0

Based on these lessons from our experience with BDM-KAT 1.0, the knowledge acquisition tool was reconceptualized and completely redesigned. The 2.0 version, while adhering to the original goals of the project: increased timeliness, support for incremental and multiple expert knowledge elicitation, and support for updating and maintenence of knowledge bases, was designed more from the perspective of the expert's and engineers it was intended to serve. In other words, the underlying theoretical model for BDM-KAT 2.0 is a model of human cognition rather than of software engineering. In addition, to address the speed and portability problems of the 1.0 version, BDM-KAT 2.0 was hosted on a SUN workstation and implemented in C.

III.1 Methodological Foundation: Models of Human Cognition

The 2.0 version of BDM-KAT differs from other tools in several critical ways. Like the 1.0 version, it has been designed and implemented to aid a trained knowledge engineer in eliciting expert knowledge about well-structured domains. Unlike its predecessor and most other tools, it is based on principles of human cognition relating to the representation and structure of knowledge in memory, yet these representations are implemented in a computationally-efficient way. The toolkit provides an environment in which the knowledge engineer and domain expert discuss, manipulate, and record the objects and relationships that provide the domain with its structure and predictability. As the elicitation progresses, a database of basic domain components is developed. Using

this database, a knowledge base and inferencing structure can be derived to operate in the selected domain.

A first premise of BDM-KAT is that all aspects of an expert's knowledge must be captured in a knowledge base for an intelligent system. Equally important is the premise that the knowledge structures as elicited and represented should be compatible with the structures used by the expert. There is substantial agreement with both of these premises within the AI and cognitive science communities (Barfield, 1986; Gammack, 1987; Hamill, 1984; Kidd, 1983; McGraw & Harbison-Briggs, 1989). However, as discussed above, the problem of eliciting and representing knowledge is highly complex. When compatibility with human knowledge structures is made a priority, it becomes necessary to address the even more perplexing issue of non-verbal knowledge. This presents an extremely difficult situation, particularly because the nature of non-verbal knowledge is not well understood by cognitive psychologists. Generally speaking, the cognitive aspects of non-verbal knowledge have been treated in more detail by Eastern psychologists than those of North America, where there is still controversy about the reality of direct visual representation (represented by the advocacy of dual coding by Paivio (1986) and its rejection by Pylyshyn (1984)). While we were reviewing the literature in cognitive psychology for results and theoretical models to support our design efforts, a former BDM consultant introduced us to Dr. L.M. Vekker, former chair of psychology at the University of Leningrad. Over the course of a 40-year career, Professor Vekker has developed a theory of human knowledge processing that integrates all levels of mental activity from perception to abstract thought. In fact, within the literature of Eastern psychologists, the work of Professor L. M. Vekker (1974, 1976, 1981) is outstanding. In a series of discussions with Dr. Vekker, we determined that his model addresses most of the points we considered necessary for developing a knowledge acquisition tool based on a cognitive model. As is summarized in the following paragraphs, his model addresses both the representation and manipulation of knowledge, provides for multi-modal representations of knowledge internally, and maintains a process-oriented view of information and reasoning.

III.1.A A Multi-modal Model of Cognition

Vekker has developed, over a research career of 40+ years, a comprehensive theory of the human cognitive system. He drew strongly on the general structure of information theory (see Billingsley (1963) for definition and discussion of the principle concepts in information theory) in building his multi-dimensional model of the human system. In particular, he defines knowledge as a special type of information within that framework and uses the principles of isomorphism and homeomorphism, or the one-to-one mapping of different functions or knowledge representation to one another to develop his model of the levels and types of knowledge within the human system and their relationships to one another. Vekker describes the internal store of information as being of two fundamentally different types (imagistic and sentential) that are reflected in multiple levels of the cognitive system. Figure 7 summarizes the types of knowledge and the increasing levels

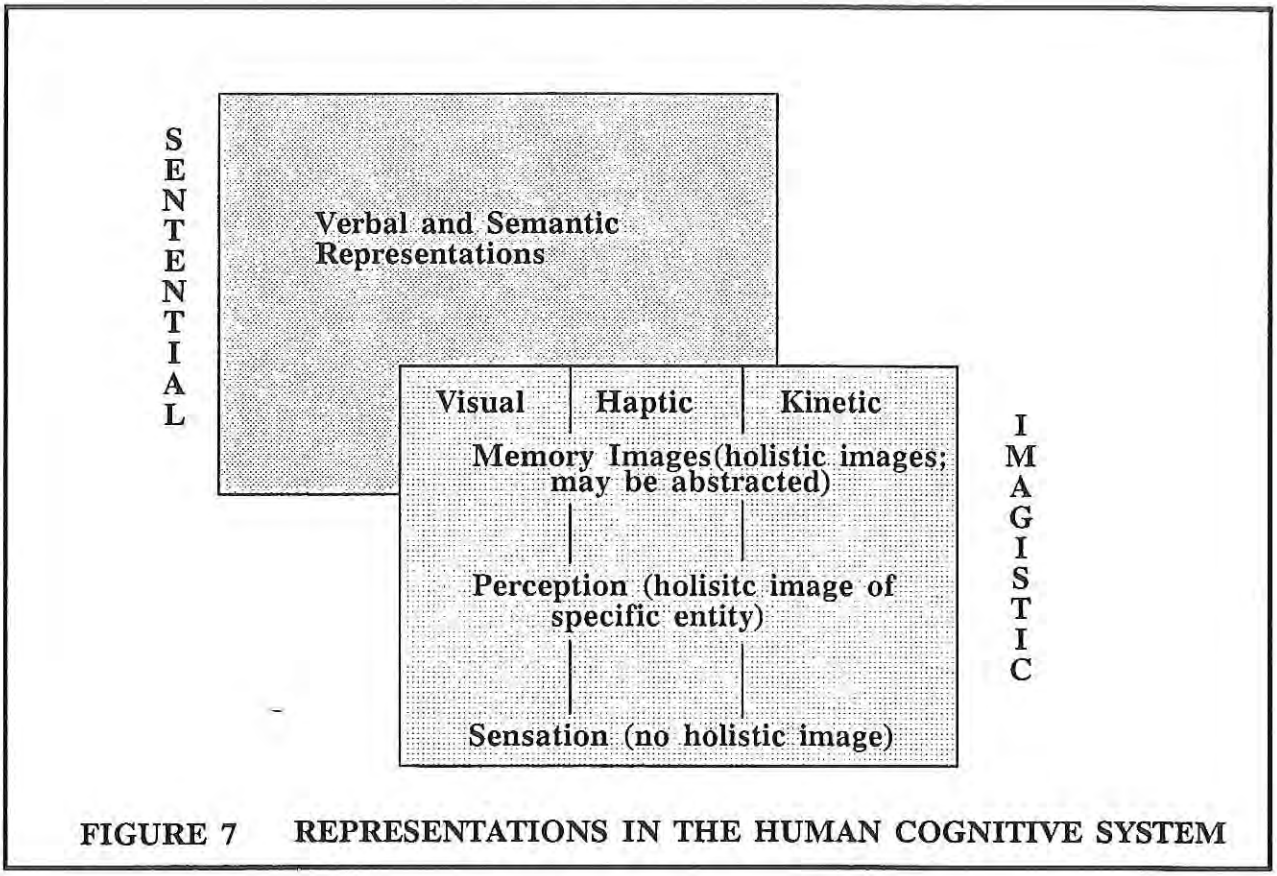


FIGURE 7 REPRESENTATIONS IN THE HUMAN COGNITIVE SYSTEM

of abstraction discussed in Vekker's model. Perceptual and memory "images" reflect some specific object or experience or a hierarchically abstracted, idealized image of a class of objects or experiences such as a prototypical object. Images include not only *visual* images but also such sensation-oriented representations as the *kinetic* sense of balance and muscle motion required to ride a bicycle and the *haptic and kinetic* awarenesses of when a football is held and thrown properly. In contrast, a sentential representation is not tied to any specific experience; it is an abstract and verbally-oriented representation. Since the imagistic components of awareness of specific and generic objects and the sentential concepts of names of such objects are represented and stored separately, thinking or reasoning processes require the manipulation of connections that exist between sentential represenations and their related imagistic representations in a process of *continuous reversible translation*, as indicated in Figure 8.

Dr. Vekker also discusses learning and representation in individual and multiple modes. While the knowledge or information necessary for reasoning in a domain is being learned, the learner will have many inconsistencies in her knowledge, particularly between imagistic and sentential representations. In contrast, experts tend to have developed stable knowledge representations that allow them to manipulate the knowledge to make inferences, to relate similar knowledge, and to be consistent in their statements, assump-

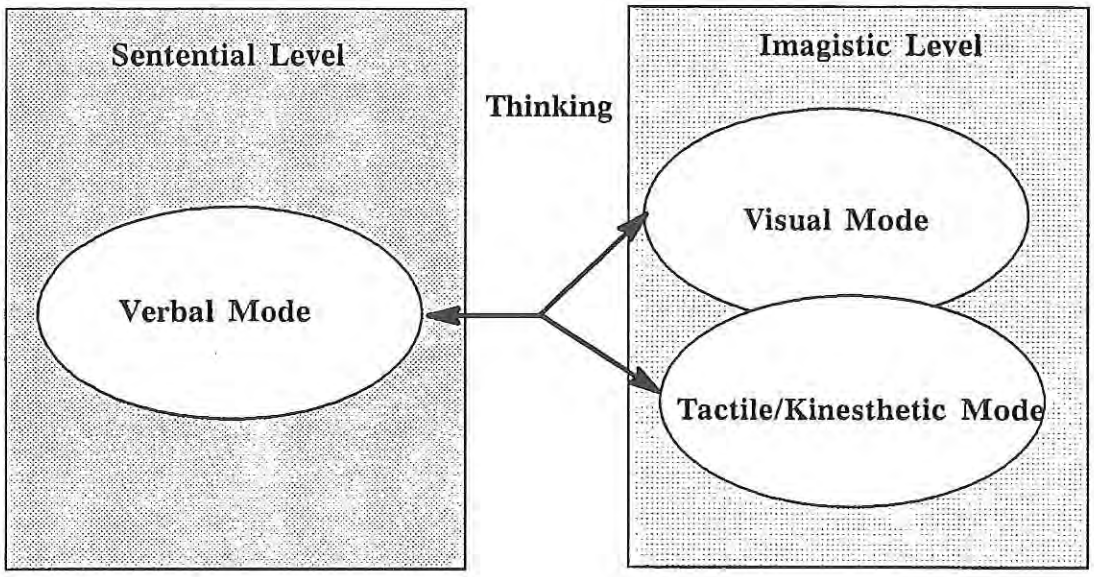


FIGURE 8 INTERACTIONS IN THE COGNITIVE SYSTEM

tions, and conclusions. The degree of instability in a student's problem solving or question answering, where manipulation of sentential and imagistic representations simultaneously is required, is a measure of the degree to which she has not yet mastered the material to be learned. Professor Vekker emphasizes in his discussions of knowledge elicitation that questions and problems must be posed to the expert that will drive the interaction of sentential and imagistic knowledge so that stability in the translations between the two "languages" can be demonstrated. Only then can the elicited knowledge be considered valid and acceptable across modes. In BDM-KAT, the methods and procedures for using the tools have been developed to encourage the expert to draw upon his/her imagistic and sentential representations to enable identification of the critical invariant relationships in the expert's knowledge. Critical invariants are those aspects of a concept or causal sequence that cannot be altered without substantively changing the nature of the concept. For example, a (American) football game has numerous characteristics, including the size of the field, shape of the goal posts, nature of the uniforms worn, and time limits on play that, if violated, do not prevent the ongoing activity from being considered a football game, albeit a "back yard game" or "pick up game" or some other informal variant. Nevertheless, if other characteristics of the game such as the shape of the ball or the means by which the ball is placed into play or transferred from one player to another, then what is played is *not* a football game at all, but another game

altogether, perhaps soccer. This exercise indicates that while those characteristics in the first list are *important* features of a game, they are not *critical invariants* of the general concept. However, if the concept is limited to an officially sanctioned inter–mural game, then those characteristics may well turn out to be critical invariants of the more limited concept.

Several characteristics of BDM–KAT are the direct result of this cognitive framework. In the current version of BDM–KAT, the interface representations have been designed to take advantage of the verbal and nonverbal aspects of the knowledge structures being elicited and the initial set of tools implemented has been selected to draw upon knowledge structures common to all human information processing. In addition, an overall integration schema for the toolkit has been partially developed that includes additional tools and a "manager" component to assist the knowledge engineer in coordinating sessions and integrating the knowledge base. The set of tools that we expect to eventually comprise the complete toolkit will reflect the cognitive system as shown in Figure 9. Briefly, the functions of the tools are as follows:

Categorizer: elicits basic domain concept structure in rigid class/subclass hierarchies and tracks concept properties.

Classifier: acts as Categorizer, but also supports creation of tangled concept hierarchies.

Time Mapper: elicits temporal sequences of events, supports decomposition of processes into event/subevent hierarchies, tracks temporal characteristics of events.

Causal Charter: elicits causally connected sequences of events, states, and concepts.

Imager: supports input of graphic images and elicitation of information implicit in the images.

In a later section of this report, the implemented tools are discussed in some detail.

III.1.B COMMON KNOWLEDGE STRUCTURES

This framework of the ways in which human knowledge is structured along with our knowledge of how such structures are organized has driven the selection and design of the initial set of tools for BDM–KAT. Expertise can rarely be elicited and represented in a single structure as humans tend to store knowledge about a domain in multiple structures. Knowledge engineers, and tools developed to assist in the process of knowledge acquisition, must be capable of eliciting, translating, and representing multiple types of knowledge structures for any given domain. Complex, knowledge–intensive domains reflect diverse, yet interrelated types of knowledge. For the initial set of tools in the BDM–KAT prototype, we have focused on those recognized to be universally applied to verbal knowledge, specifically categorical, temporal, and causal structures. The following

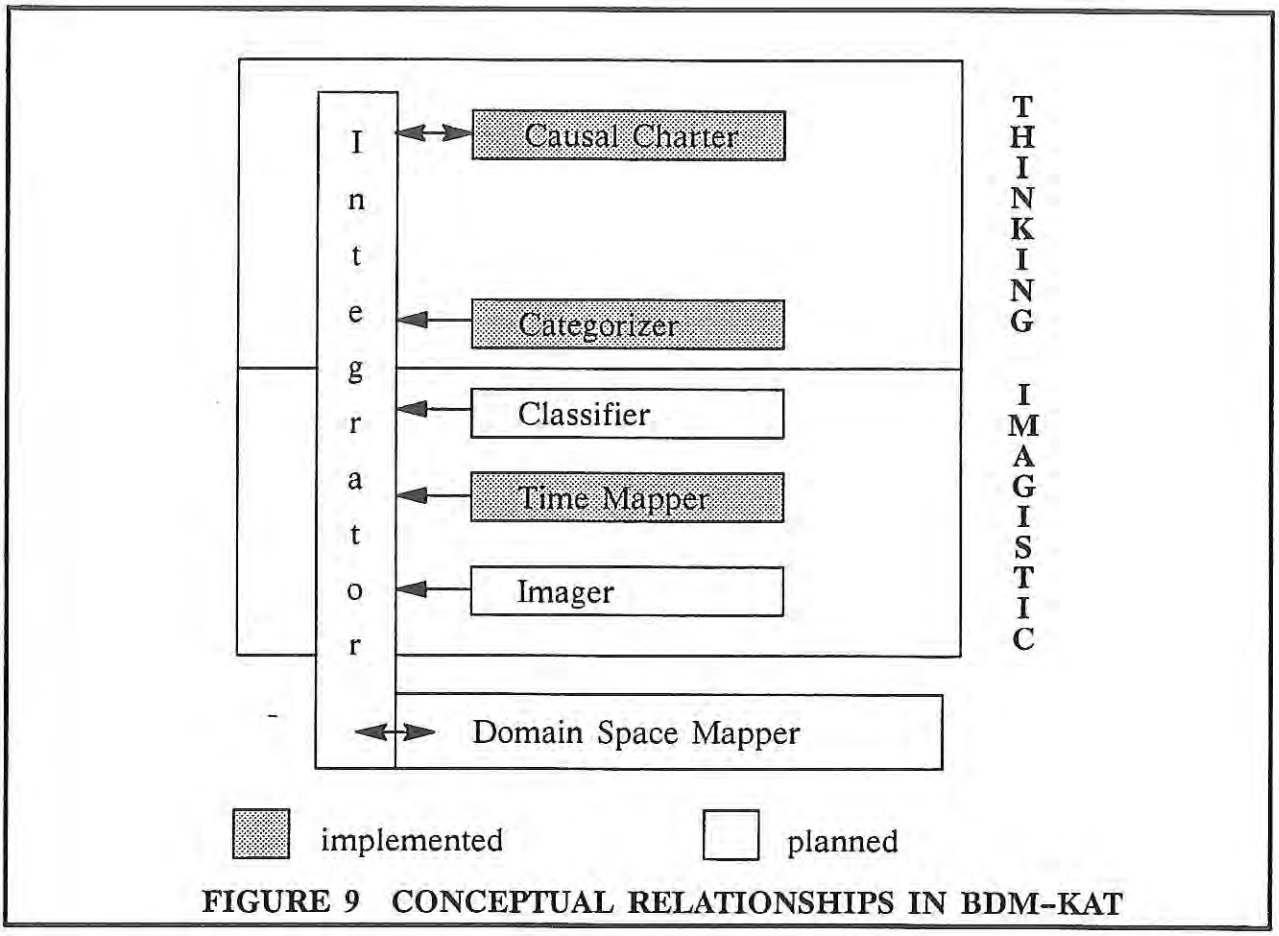


FIGURE 9 CONCEPTUAL RELATIONSHIPS IN BDM-KAT

paragraphs present basic information about the nature and support for the generality of these structures.

Categorical Structure of Knowledge

People tend to divide the world into relatively neat categorical structures. While for laymen the categorical divisions are frequently unstable (e.g., individual items may change their positions in terms of both category membership and typicality within a category (Barsalou, 1986)), experts in a specific field tend to hold stable and shared categorizations of their domain (Rosche, et. al. 1975). A substantial portion of one's understanding of a domain rests in knowledge about the items that appear in the domain, their individual and collective characteristics, and the relationships between them. Consider again, for example, our football game and its position in a general categorical structure of games. Few (even non-experts) would have trouble producing a taxonomy such as that shown in Figure 10 and naming the characteristic properties of each node. Undoubtedly, a sports professional would quickly generate a more detailed, and possibly differently structured taxonomy. Because this organization is natural to the domain expert, knowledge elicited within this structure is likely to be more easily obtained, more easily modified, and, once accepted by the expert, more accurate.

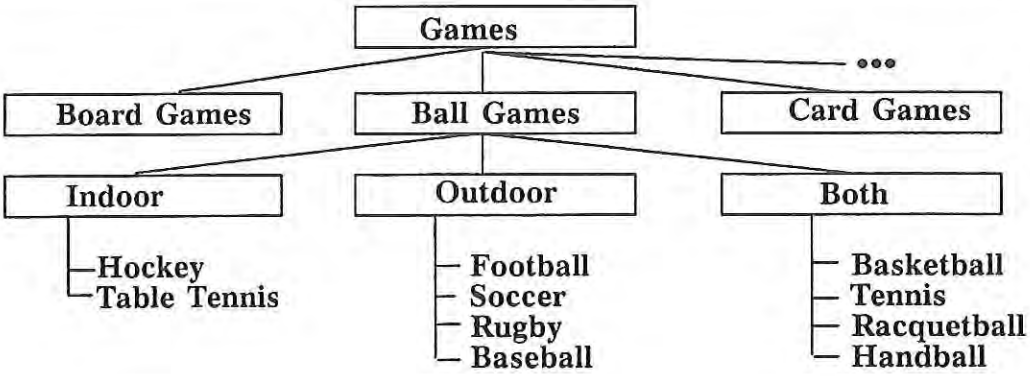


FIGURE 10 SIMPLE TAXONOMY OF GAMES

Temporal Organization of Events

In addition to dividing objects in the world into categorical sets, people structure their knowledge and memories of events into orderly sequences. Autobiographical knowledge appears to be organized internally into "mental time line models" in which the temporal relationships between episodes are specified (Barsalou, 1986). Similarly, individual events and generalizations of events appear to be stored menally as ordered sequences of actions and/or decisions, such as those in the familiar restaurant script, doctor script, and so forth (Mandler, 1979; Nelson, 1978; Schank & Abelson, 1977). Where the episodes or process events can be decomposed to enabling or sub–events, hierarchical relationships exist between them similar to those between levels in a categorical taxonomy. This structural nature of event memory can be exploited in eliciting knowledge about processes that occur over time. Drawing from this model the domain expert can tap and share the order, temporal characteristics and relationships, and hierarchical associations between steps in the process.

Causal Structures in Knowledge

Finally, experts can be partially defined as those individuals who understand the *why* and *how* of a particular domain. All people, even novices, construct causal mental models of the world around them (Kelly, 1955). However, causal models held by a novice may reflect incorrect assumptions or focus (e.g., correlations), while those held by an expert tend to be more accurate. Consider, for example, the knowledge used by auto mechanics. While they do develop, with expertise, extensive classifications of diagnoses by symptom sets, they continue to solve most diagnostic problems by tracing the presented malfunction through its causal antecedents to (hopefully) a single cause. The networks of causal connections supporting this reasoning have been shown to improve in specifiable ways with increasing experience and training (Lancaster & Kolodner, 1987;

1988). As with categorical and temporal organizations of knowledge, causal models can be elicited directly from an expert--here by focusing interactions on the expert's explanations for states and conditions in the domain of interest. Of these three types of knowledge, causal models are the most similar to the rules usually sought during the knowledge acquisition phase of knowledge-based system development.

III.2 Design And Implementation

BDM-KAT is a toolkit that has been designed to enable a knowledge engineer to elicit knowledge based on both the structure and modality in which it is most likely to be represented in the domain expert's internal cognitive model. To accomplish this, BDM-KAT consists of three cooperating knowledge engineering tools: Categorizer, Time Mapper, and Causal Charter. Their high-level structure is shown in Figure 11. *Categorizer* elicits knowledge about the stable, underlying structure of a scientific domain via the

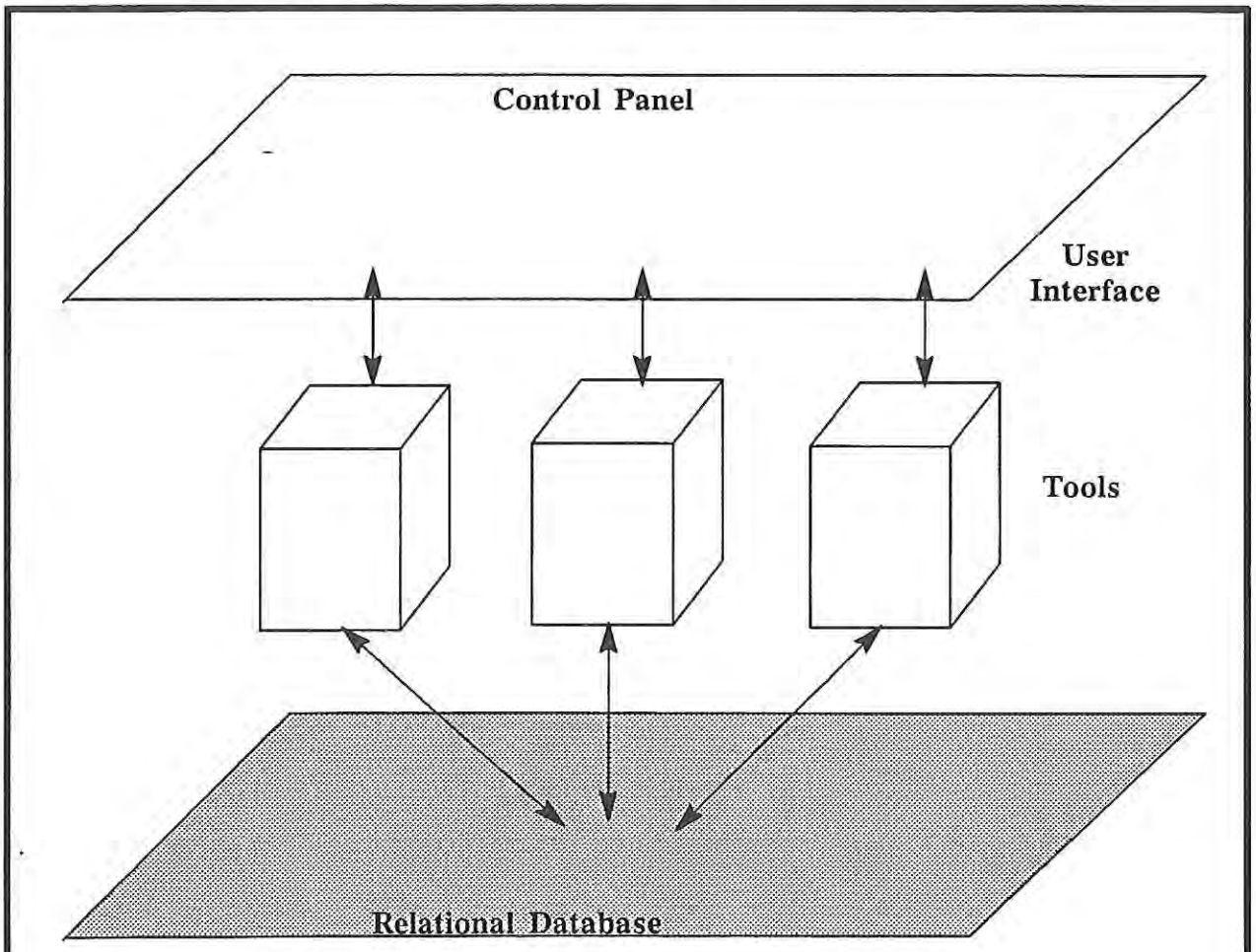


FIGURE 11 CURRENT STRUCTURE OF BDM-KAT

drawing and "filling in" of a tree-like, taxonomic structure. *Time Mapper* provides a context in which the domain expert positions items on time lines to express knowledge about the temporal characteristics of, and relationships between, events and subevents within a process. The knowledge engineer uses *Causal Charter* to elicit cause-effect knowledge for domain functions and events. Each tool provides the knowledge engineer and domain expert with a means for 1) focusing attention on the aspect of the domain being elicited, 2) recording information as it is elicited, and 3) verifying the information as it is elicited.

Each tool uses textual and graphic interface representations, and all tools transfer information to, and interact with, a single database management system (DBMS). The final database representation of the knowledge has three potential uses. First and foremost, the database should be of form and content sufficient to serve as an intact knowledge base for insertion into a knowledge-based system. Secondly, the structure of the database itself should provide significant support for the generation of knowledge documents, which are on-line or hard-copy structured summaries intended for review, verification, and validation activities by the domain expert, knowledge engineer, and other system development team members. Finally, because the knowledge will be stored in a supported DBMS, it will be possible for human users of the knowledge to query and browse the content of the database for assistance in decision making outside of the context for which the knowledge-based system has been developed.

All tools are accessed from a common control panel displaying icons for each tool and a set of three buttons. The buttons can be used to generate prompt screen dumps of the current session, open an on-line notepad to capture ongoing dialogue that may not be immediately applicable to the session, and terminate the session. The tools are opened by a single mouse click on the appropriate icon. All three tools use basically the same interface as screen dumps in later sections of this paper will demonstrate. The left side of the window contains the active canvas in which objects are created and their characteristics and relationships are specified. Along the right hand side of the canvas are tool and object management buttons.

The following sections briefly describe the individual tools. This version of BDM-KAT has been successfully demonstrated on several occasions and has been used to elicit a significant data base of HIP information.

III.2.A Categorizer

Categorizer elicits the underlying structure of a domain by requiring the expert to construct a taxonomy or set of taxonomies. As discussed earlier, categorical knowledge is commonly held for any domain or topic area. Furthermore, it is critical for constructing a domain model because the properties of specific objects and object classes will dictate much about how the knowledge can be used.

Using Categorizer the expert initially creates and names objects that exist in the taxonomy and places them at the appropriate position within the structure. The expert

typically works in a top-down fashion, but the tool does not restrict the level at which objects can be named. Figure 12 shows a completed HIP taxonomy in Categorizer. After the taxonomy has been developed the essential, contingent, and second-order properties of each domain object are elicited from the expert. (A clear knowledge of object properties is necessary for the success of the overall knowledge engineering effort because it will be these properties that will determine how the model can be perturbed during future verification activities.) Object properties are best elicited in tandem with object taxonomies for two primary reasons. First, this sequence follows from the structure of the taxonomy itself. Second, while the expert's attention is focused on the taxonomy, s/he will be less likely to make errors in identifying properties for individual objects.

Categorizer creates and builds within the database a series of entities and relationships between them. The domain objects and properties specified during elicitation are represented individually in the database as are their relationships.

III.2.B Time Mapper

Whereas Categorizer elicits taxonomic and part-whole relationships among objects, Time Mapper elicits the structure of events or actions that occur during the execution of a process. Specifically, Time Mapper elicits the hierarchical relationships among events and subevents (at multiple levels), the overall duration of each event, and the specific starting and ending times of each subevent within an event. This elicitation effort is critical to developing a full process model because it provides both the decomposition

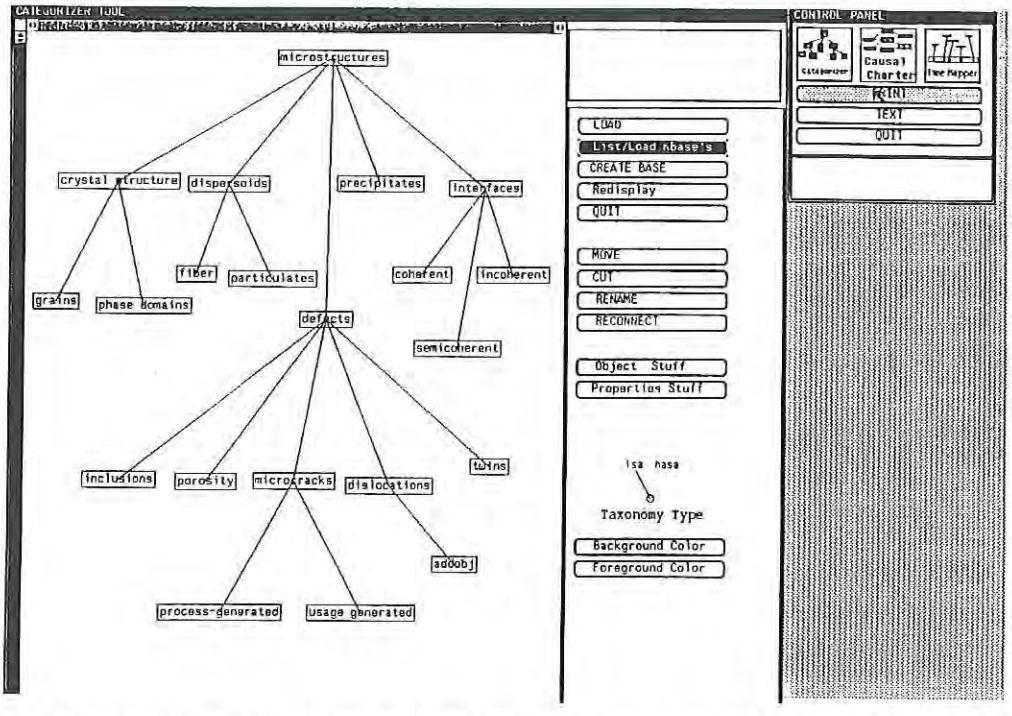


FIGURE 12 PARTIALLY COMPLETED TAXONOMY IN CATEGORIZER

of the process into well specified components and the temporal relationships necessary for control. The expert is asked to name the process as a single event and then to name the subordinates of that event. This cycle iterates with increasing specificity, such that the subevents at one level are considered as events at the next level, until the events can be decomposed no further. Figure 13 shows a the process execution steps for a HIP cycle in Time Mapper.

III.2.C Causal Charter

The third tool, Causal Charter, is used to elicit high level "rules" and low level cause-effect chains. The rules are intrinsically implicit and are supplied by the expert through a description of an item to be "charted". The item under consideration may be either a function of a domain object (specified during Categorizer as a functional property of that object) or a subevent from the process decomposition state of Time Mapper. The structural elements in the item description are separated from other specified functions and are grouped together to indicate those elements that support each function in the description. Thus, the generation of clusters of structural elements (as units) are related to specific functions. The expert is asked to select those items from the description that provide support for each function given in the description. When specifying the cause-effect chains the domain expert selects the elements from the mechanism *in the order in which they are caused* or required by the mechanism. Thus a completed chain indicates the process by which the final state or element is reached. Causal Charter can also serve as a

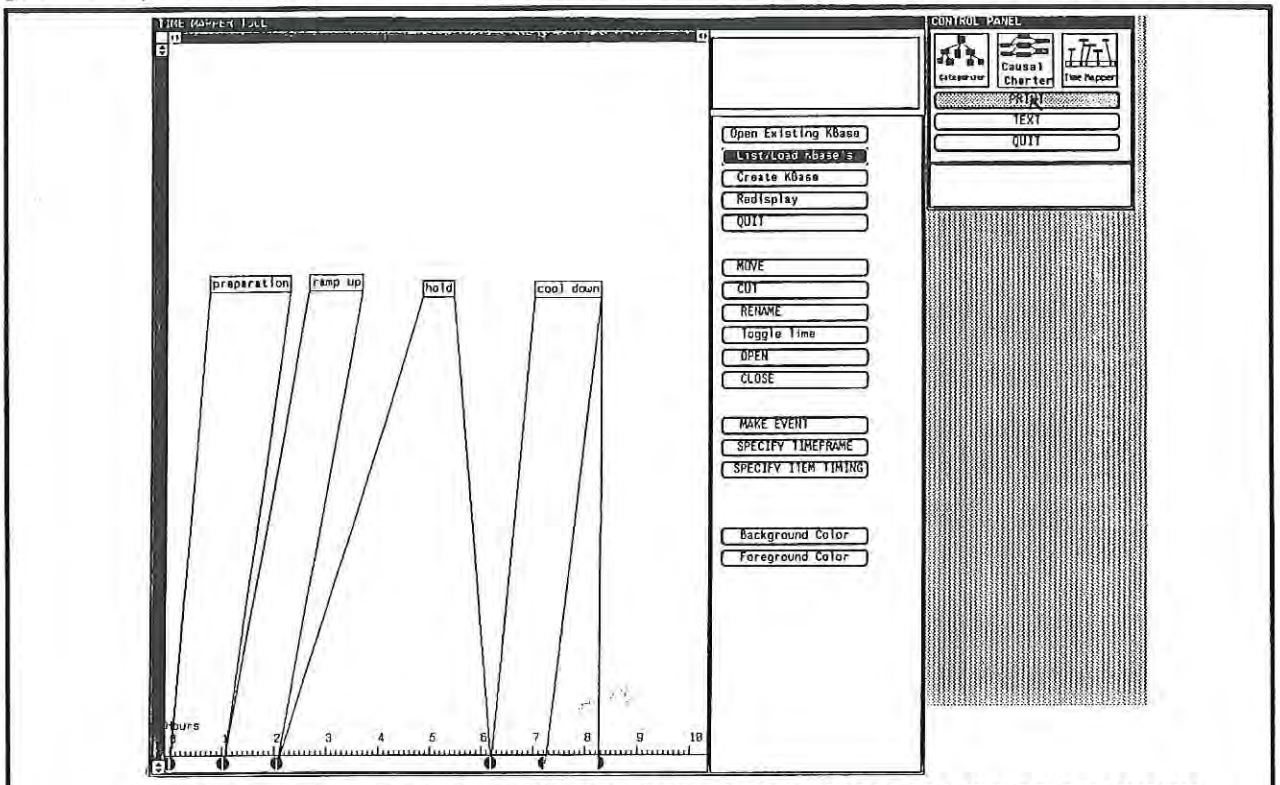


FIGURE 13 PARTIAL PROCESS SPECIFICATION IN TIME MAPPER

verification mechanism on some components of the categorical and process knowledge elicited previously.

III.3 BDM-KAT: An Example of Its Use

Given this description of the general functionality of each tool, let us now consider their coordinated use in a knowledge acquisition effort. As with any such exercise, the knowledge engineer's first task is an organizational one; in this case identifying the types of knowledge to be elicited and selecting the tool to support each session. For example, suppose a knowledge engineer were tasked to structure a knowledge base for a software engineering support tool. Analysis of the task domain would allow the knowledge engineer to determine that the tool will need to have two primary forms of knowledge in its knowledge base: that about the timing and sequence of various planning, documentation, code development and testing activities leading up to a final software product; and that about the set of documents and document components required for the developing and final software product, their characteristics, and their relationships to one another and the steps in the process. From this characterization of the knowledge base requirements, the knowledge engineer concludes that the first of these classes of knowledge can be elicited using the Time Mapper tool and the second using the Categorizer tool. Because the process of software development is only tangentially driven by direct causal factors the knowledge engineer makes no initial plans to use the Causal Charter tool during eliciation activities.

Based on preliminary conversation and scheduling with the domain expert, the knowledge engineer decides to begin the elicitation activities with a series of sessions using the Time Mapper tool. This decision is driven by the domain expert's statement that s/he will find it easier to identify all the document requirements from their relevant points in the process and sense that the process sequence may be the more straightforward to express, at least in an idealized form. In the first session, the expert and knowledge engineer focus on establishing the overall time frame (months or years) for developing a large-scale software product and on establishing the general stages through which such a project should move. During discussions of timing factors, the knowledge engineer focuses the domain expert on how much time is allocated to each phase of the process and how much they can or should be allowed to overlap one another. In subsequent sessions, each phase of the process is broken down into smaller steps and the timing and sequencing of the steps discussed until the domain expert states the the tasks are defined as finely as possible or as finely as is appropriate for consideration by the support tool. At the end of each session, the domain expert is provided with a hard copy of all the material discussed during the session, and all material discussed at previous sessions to support review and correction of already elicited material and the expert's own preparation for the next session. In particular, the expert is asked to prepare for subsequent sessions by considering the next level of task breakdown.

With the process sequence and timing structured to the expert's satisfaction, the knowledge engineer is ready to turn to a series of sessions using the Categorizer tool. In preparation for the first session, the domain expert is asked to review the process sequence and to note down document requirements that are linked to specific points in the process. In addition, the domain expert is asked to review or to bring any support materials that define the structures of the documents insofar as they can be separated into independently prepared components. Armed with this information, the domain expert and knowledge engineer begin their first session with the Categorizer tool by enumerating the set of documents required and placing them into a hierarchical structure that, at its highest level, reflects the general phases of the development process.[1] For each document, the domain expert is asked to specify the properties of the document. Properties, in this domain would include such things as the purpose of the document and any official numbering conventions for document identification. With the set of full documents defined and described, the expert and knowledge engineer then proceed to address the lower level sections and subsections of each document, placing them into their appropriate relationships to the parent documents and specifying their characteristic properties until the documents have been defined to the finest level at which a section could be generated somewhat independently of other sections.

This brief review of the use of BDM-KAT has demonstrated that the tools constructed map well into the types of knowledge required by knowledge based systems of several types. In addition, the knowledge engineer and expert are aided in both preparation for and execution of elicitation sessions by their knowledge of how the tool will be used to structure and focus the discussion for each session and by the easy availabilty of hard copies of the material for review and modification purposes.

III.4 BDM-KAT Performance in the HIP Domain

The Categorizer and Time Mapper tools were used in a series of knowledge engineering sessions with four HIP domain experts over the period March-June 1989. These sessions generated a set of HIP data bases providing a structured view of both the overall processing constraints and the microstructures, mechanical properties, and usability factors of HIPped materials. The overall plan for the elicitation effort was specified in the HIP Knowledge Engineering Plan.

As in the hypothetical example covered in the previous section, knowledge engineering for HIP began with domain familiarization, in this case a months-long informal process, and some high-level domain structuring. From this, we derived a set of domain categories in which knowledge would be required for the ultimate functioning of a HIP knowledge-based controller. Given the domain information discussed in Section I and

[1] While, in its current implementation, BDM-KAT does not directly link the Categorizer information to the events defined in Time Mapper, having the process structure available during the Categorizer elicitation does assist the knowledge engineer and expert in defining the hierarchical relationships and, in BDM-KAT's final form, the connections would be generated and maintained in the knowledge base.

other design information from the HIP program, we determined that knowledge was required specifying the temporal aspects of the HIP process in its entirety; defining the microstructure and mechanical properties of titanium aluminides (TiAl), the material under consideration; and specifying the selection parameters for materials to be HIPped in general. Given this overview of the knowledge engineering requirements, sessions were scheduled with appropriate experts. Because we were initially pursuing knowledge in separable aspects of the domain, knowledge engineering with each expert progressed independently. In the following sections, we present the progress and results of the knowledge engineering efforts between January and April 1989.

III.4.A Temporal Considerations

Knowledge engineering began with two sessions in January 1989 with H.N.G.Wadley as expert using the Time Mapper tool to derive a full time map of a specific HIP process from the point of material selection up to the point at which the final part is inspected and shipped. Because our main concern was with the sequencing of actions and possible areas of overlap. the time map was structured as if work were done continuously with no nights or weekends. With the tool, Dr. Wadley was able to break down the overall process in two one–hour sessions. During the first session, the discussion focused on specifying all events at the most general level and Figure 14 shows a screen dump of the results of that session. The subsequent session focused on specifying more detailed breakdowns of events for all events preceeding the setup and execution of the HIP run

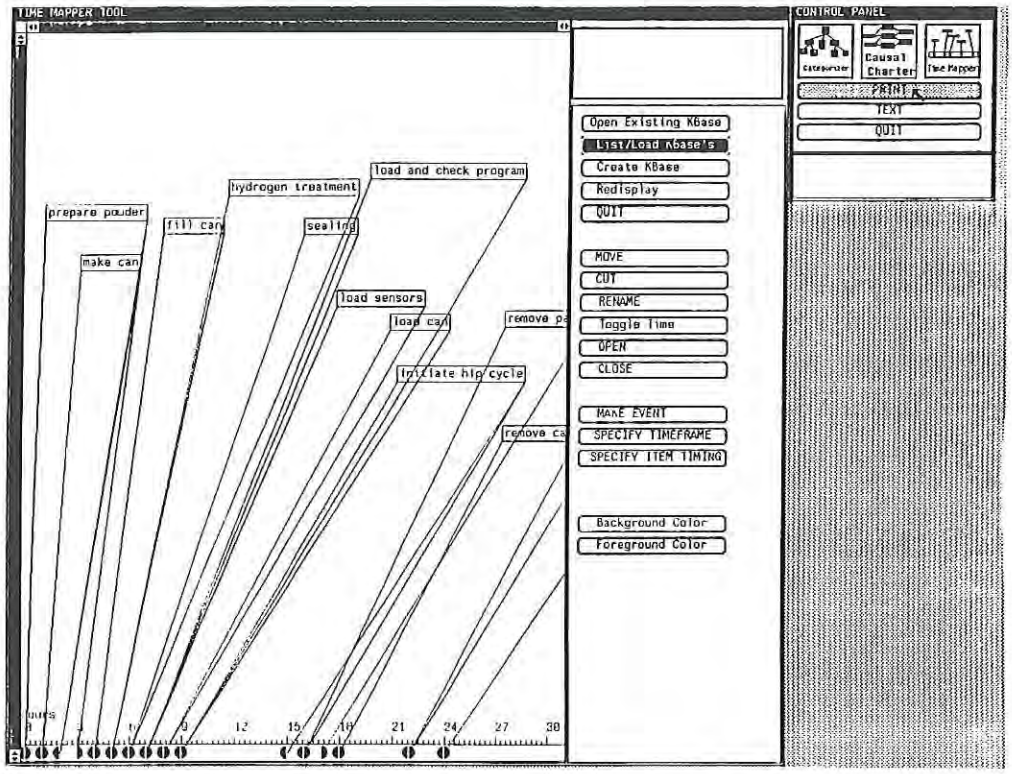


FIGURE 14 TOP LEVEL OF HIP TIME MAP

and for the final few steps. The detailed breakdown of events occurring within the HIP run itself was obtained during a one hour session with Dr. R. Schaeffer April 13, 1989. Figure 15 shows an example of the most detailed level in the elicited HIP timemap. A

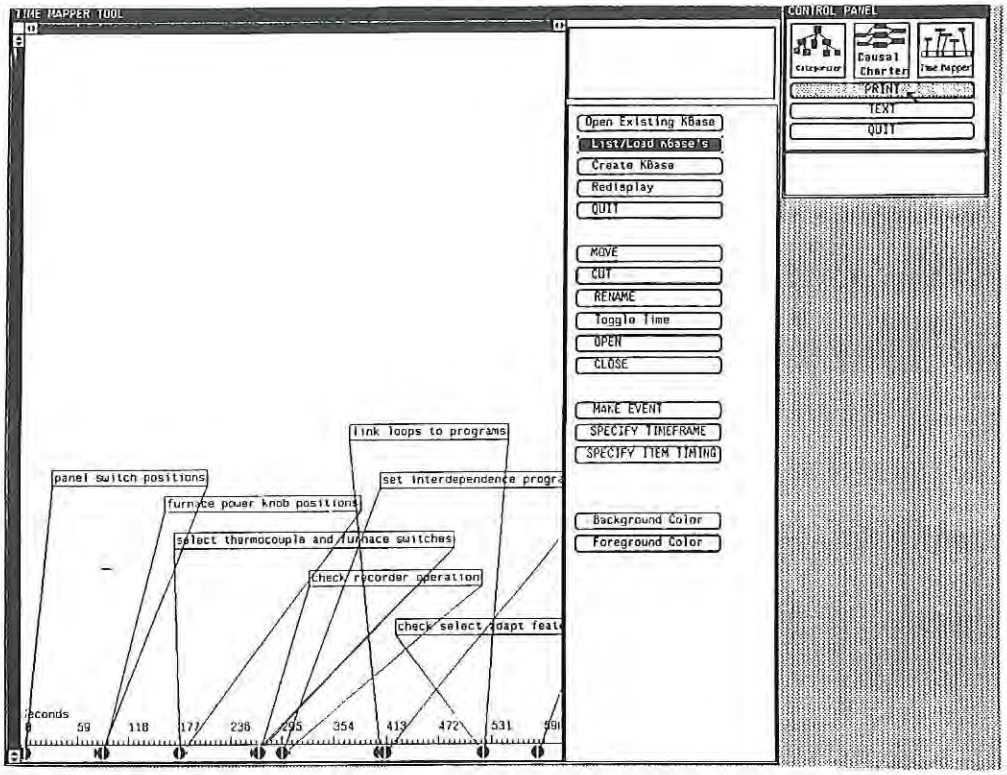


FIGURE 15 SAMPLE PORTION OF FINE LEVEL HIP TIME MAP

final session with Dr. Wadley to review the inputs of Dr. Schaeffer was held at the end of March 1989. In this elicitation process, the two experts consulted were consistently in agreement with one another about both the specific events to be included and their temporal relationships.

III.4.B Mechanical Properties and Selection Criteria

An initial categorization of the mechanical properties of advanced materials and the defining characteristics of those properties was developed during a one hour session with Dr. Roger Clough on April 13, 1989. The purpose of this session was to structure information about strength and usability of materials after HIPping and the factors that influence those characteristics. Figure 16 shows the overall categorical structure that was elicited. During a second one-hour session on April 19, 1989 Dr. Clough defined the characteristic features of each item within the categorical structure of mechanical properties. Figure 17 shows an sample screen from that session. During the same session, an initial structuring of the parameters that affect selection of a material for use in a specific part was developed. Creation of this structure consumed approximately half of the one-hour session. Figure 18 shows the results of the discussion.

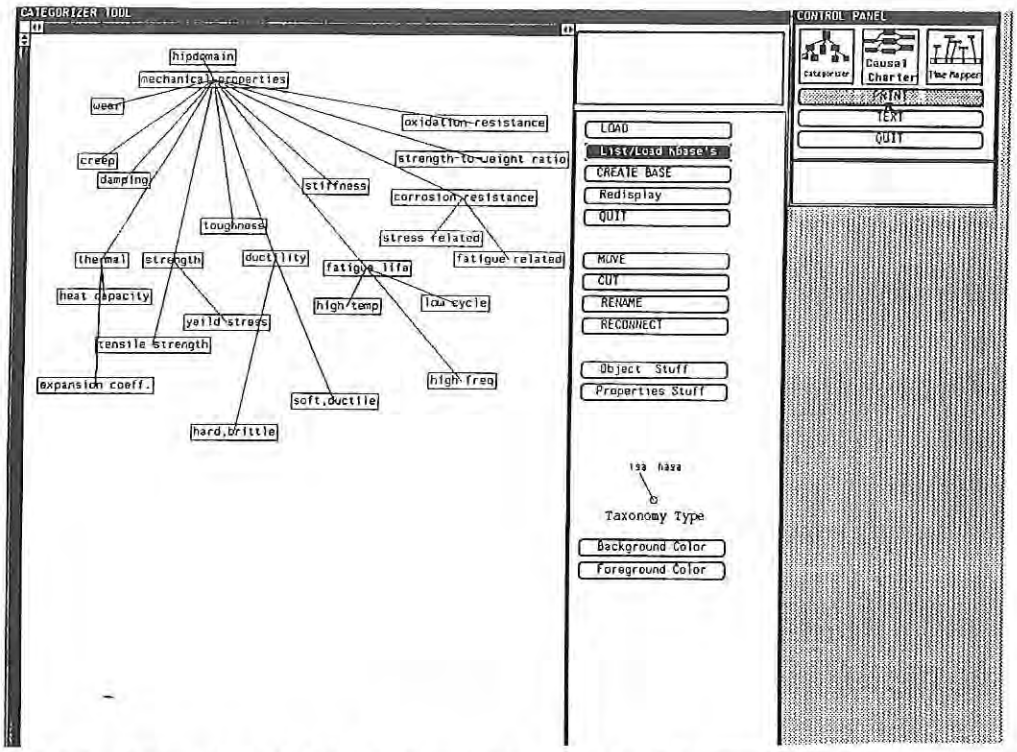


FIGURE 16 CATEGORICAL STRUCTURE FOR MATERIAL PROPERTIES

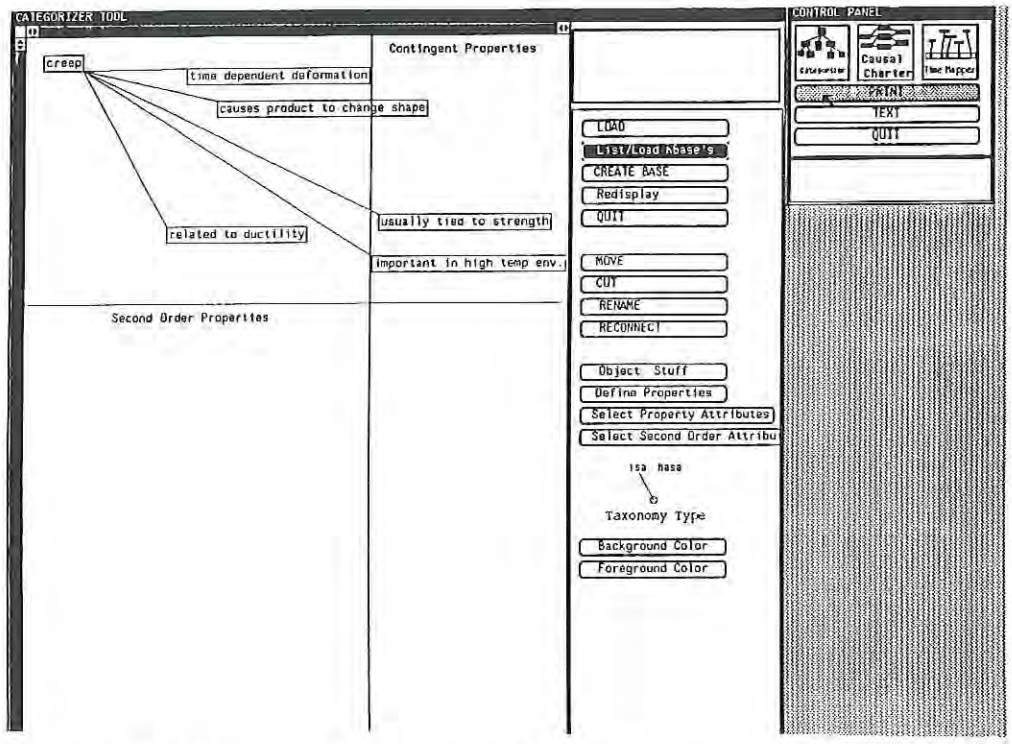


FIGURE 17 CHARACTERISTIC FEATURES OF CREEP

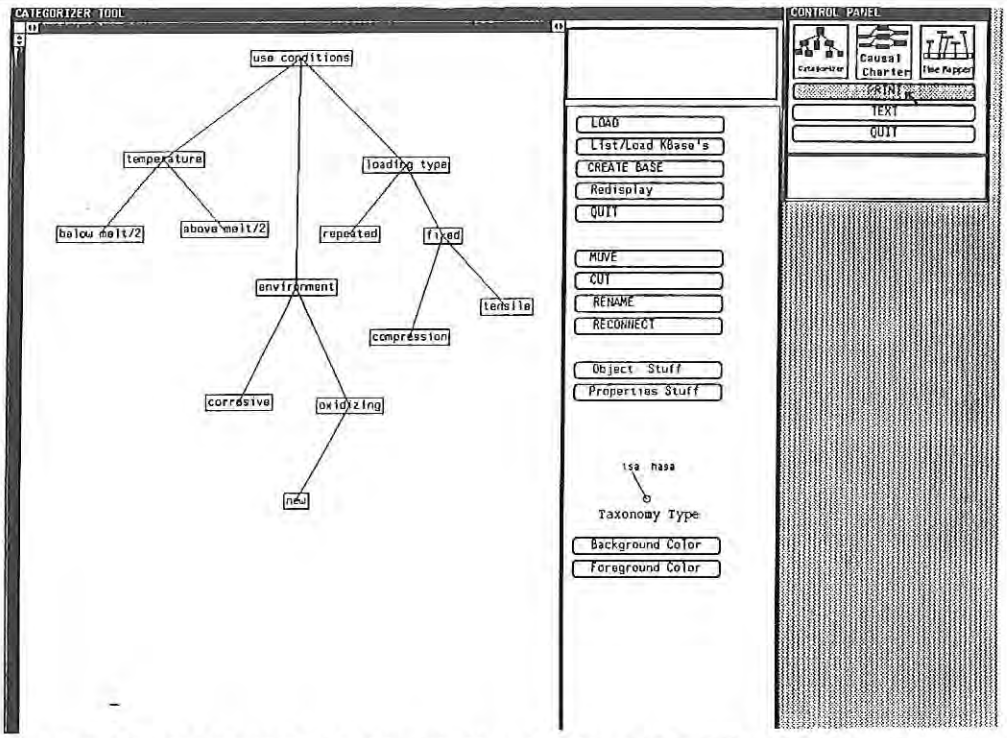


FIGURE 18 CATEGORICAL STRUCTURE FOR USE SELECTION

III.4.C Material Microstructures

Two one-hour sessions, on April 13 and 19, 1989 were held with Dr. Richard Fields to discuss the microstructures of advanced materials and their characteristic properties. In the initial session, Dr. Fields named the important microstructures, but encountered severe difficulty placing them into an organized categorical structure. Figure 19 shows the structure that resulted from this first session. As can be seen, the major concepts are present, but some confusion is present between subcategories and characteristic properties of the concepts. Dr. Fields was provided with a copy of this screen and asked to consider the relevant knowledge during the week between sessions. During the second session, a more well-structured representation was developed and characteristic properties were identified for all concepts named within less than one hour. This structure was shown in Figure 12.

III.4.D Failure Modes

During June 1989, an additional series of knowledge engineering sessions focused on characterizing possible failures during execution of a HIP cycle were held with Mr. John Wlassich of BDM. This effort required a series of three 45-minute sessions and yielded the overall structure shown in Figure 20 and characteristic property definitions for all concepts such as are shown in Figure 21.

IV. OVERALL RESULTS AND IMPLICATIONS

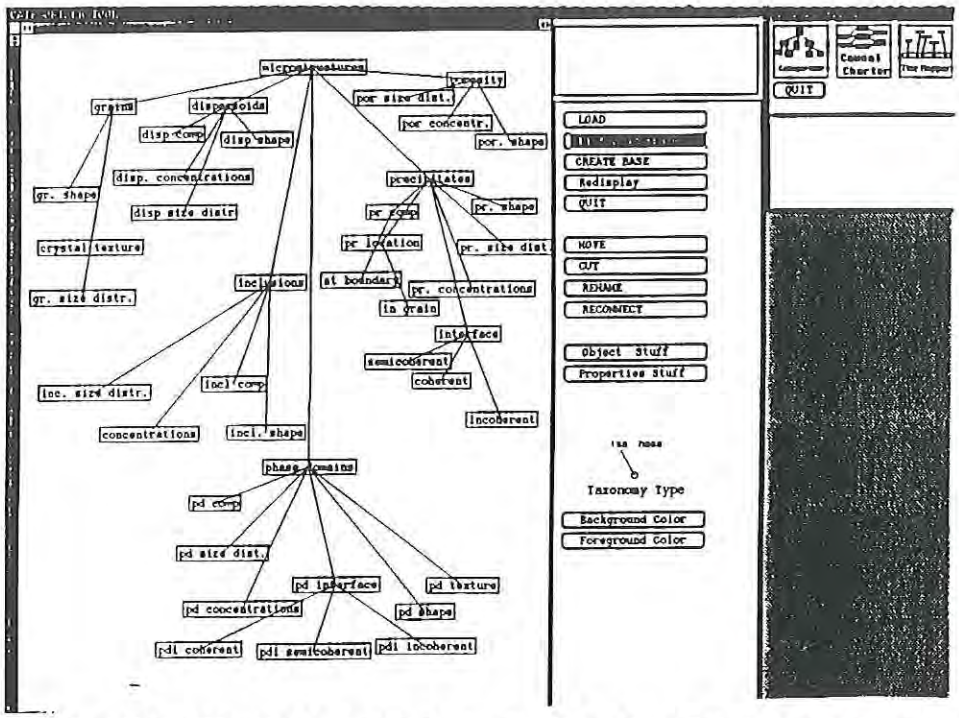


FIGURE 19 CATEGORICAL STRUCTURE OF MATERIAL MICROSTRUCTURES

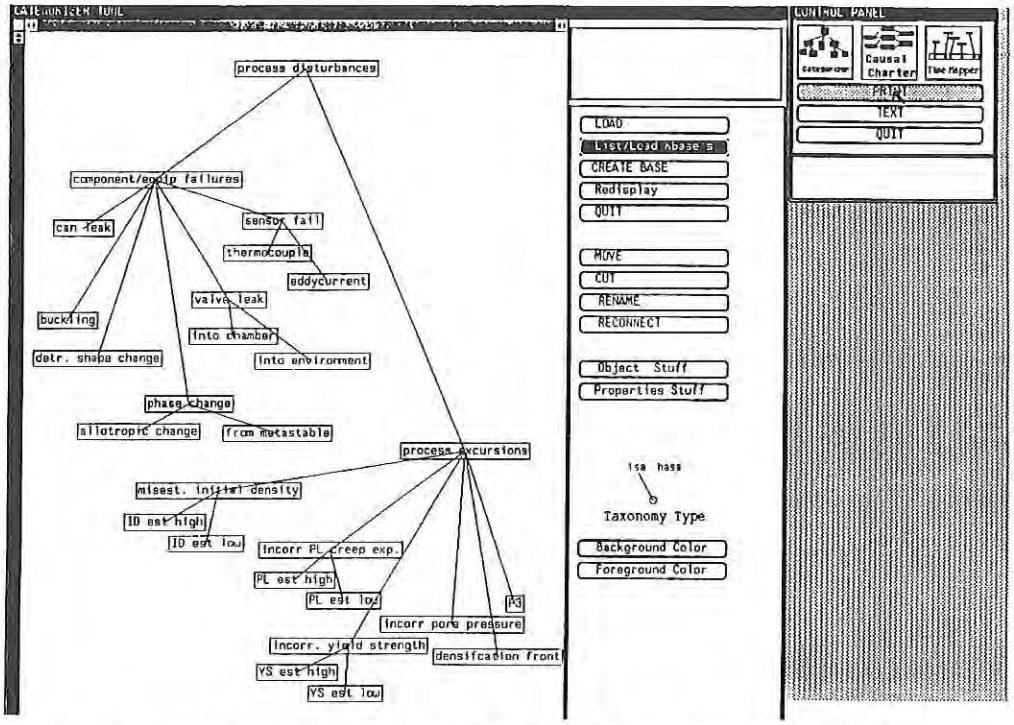


FIGURE 20 CATEGORICAL STRUCTURE FOR FAILURE MODES

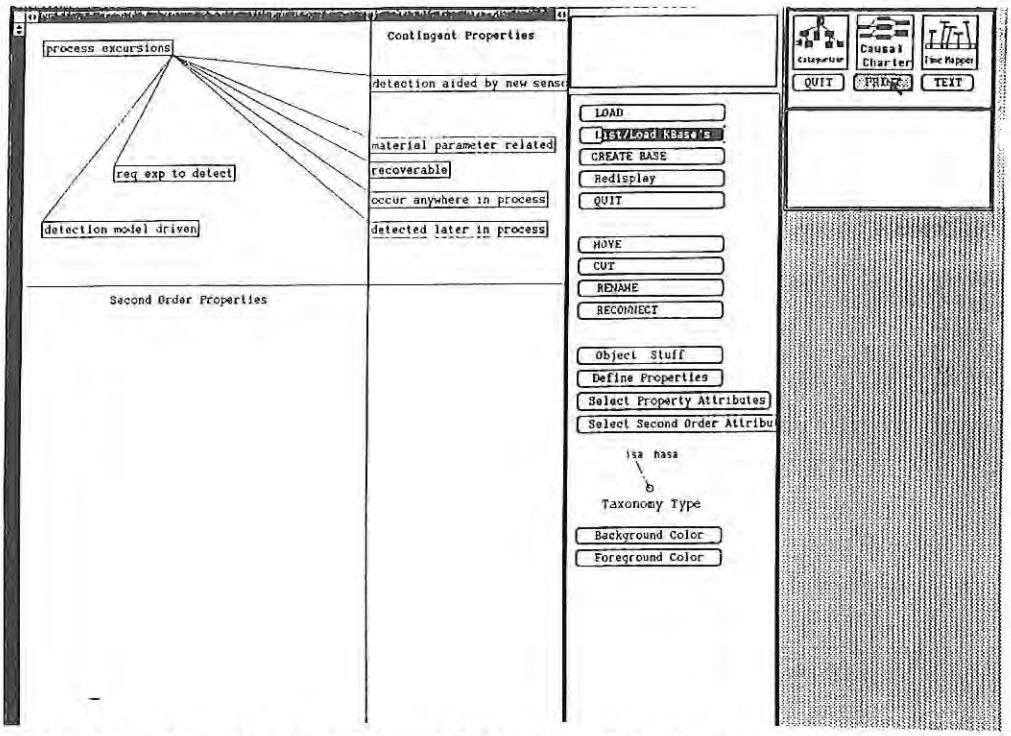


FIGURE 21 SAMPLE OF CHARACTERISTIC PROPERTIES FOR FAILURES

BDM–KAT is a cognitively–based toolkit that represents a major shift in the development of tools to support the knowledge acquisition process. While previous tools assisted the knowledge engineer in efforts related to programming or implementation, BDM–KAT has been developed based as an application of cognitive research to assist the knowledge engineer in eliciting and representing knowledge in a cognitively valid, computationally efficient manner. We believe BDM–KAT has the potential to provide significant assistance to knowledge engineers in three broad areas. First, BDM–KAT can streamline the overall process by:

- representing objects at the appropriate conceptual level;
- enabling elicitation and representation of fine–grained temporal knowledge;
- enabling direct elicitation of causal knowledge;
- capturing relevant heuristic knowledge provided during other sessions on–line; and
- using a high–quality, friendly user interface.

In addition, BDM–KAT will improve the nature of the domain expert's interactions during knowledge elicitation by:

- providing a high–quality interface;
- supporting verbalization of knowledge by the expert via cognitive compatibility;

- supporting more focused and efficient sessions; and
- supporting immediate print-out and review of elicited knowledge.

Finally, BDM-KAT enhances the verifiability of the elicited knowledge by:

- supporting immediate print-out and review of elicited knowledge;
- allowing on-line browsing of all elicited knowledge at any stage of elicitation;
- using cognitively compatible interface representations; and
- reducing the amount and number of "translations" between the expert and the knowledge base.

In addition to enhancing the quality and timeliness of the knowledge acquisition process, BDM-KAT will generate a knowledge base whose contents reflect the expert's mental organization. Such a structure is most likely to result in a knowledge-based system, intelligent tutoring system, or decision aiding system that can be used effectively in operational settings. It is our position that the cognitive foundations of BDM-KAT and its orientation toward the acquisition of knowledge in a form familiar to and comfortable for the domain expert have resulted in a system whose utility is not constrained to a specific domain or domain-class.

The overall concept of BDM-KAT calls for a fully integrated set of tools and a coherent, integrated database representation of the elicited knowledge to which inferencing techniques can be applied. Research on knowledge acquisition in the context of *BDM-KAT* development is continuing with the emphasis moving toward developing the capabilities that will realize the toolkit's potential for general purpose use in three broad technical areas of research:

- visualization of knowledge for elicitation and control purposes;
- integration of the elicited knowledge and the BDM-KAT tools; and
- interfaces between the environment and its users.

These topics will be pursued and new tools and capabilities implemented as the appropriate designs are developed. In all continuing work, the predictions and parameters of the human cognitive systems, as modeled in the multi-dimensional model discussed earlier will be used to define boundaries and requirements of the system.

REFERENCES

Barfield, W. (1985). Expert–novice differences for software: Implications for problem–solving and knowledge acquisition. *Behavior and Information Technology, 5(1).*

Barsalou, L.W. (1986). The content and organization of autobiographical memories (unpublished draft).

Billingsley, P. (1963). *Ergotic Theory and Information.* J. Wiley & Sons: New York.

Blaxton, T.A. & L. Reeker (1988) A tool for knowledge acquisition in process domains.

Gammick, J.G. (1987). Modelling expert knowledge using cognitively compatible structures. In: *Proceedings of the Third International Expert Systems Conference*: Oxford.

Hamill, B.W. (1984). Psychological issues in the design of expert systems. In: *Proceedings of the 28th Annual Meeting of the Human Factors Society*: San Antonio, TX.

Hayes–Roth, F., D. Waterman, & D. Lenat (1983). *Building Expert Systems* Addison–Wesley.

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

Kidd, A.L. (1983). Human Factors in expert systems. In K. Coombes (ed.) *Proceedings of the Ergonomic Society Conference:* Taylor and Francis, London.

Kushner, B.G., R.A. Geesey, P.A. Parrish, & S.G. Wax (1987). A knowledge acquisition tool for the intelligent processing of materials. In R.J. Harrison & L.D. Roth (eds) *Artificial Intelligence Appliciations in Materials Science.* The Metallurgical Society, Inc.:Warrendale, PA.

Lancaster, J.S. and J. L. Kolodner. (1988). Varieties of learning from problem solving experience. In: *Proceedings of the Tenth Annual Conference of the Cognitive Science Society.*

Lancaster, J.S. and J. L. Kolodner. (1987). Problem solving in a natural task as a function of experience. In: *Proceedings of the Ninth Annual Conference of the Cognitive Science Society.*

Mandler, J. (1979) Categorical and schematic organization in memory. In C.R. Puff (ed) *Memory Organization and Structure.* New York: Academic Press.

McGraw, K. and K. Harbison–Briggs (1989). *Knowledge Acquisition: Principles and Guidelines.* Englewood Cliffs, NJ: Prentice–Hall.

Nelson, K. (1978). How children represent knowledge of their world in and out of language: A preliminary report. In R.S. Siegler (ed) *Children's Thinking: What Develops?.* Hillsdale, NJ: Lawrence Earlbaum, Associates.

Paivio, A. (1986). *Mental Representations: A Dual Coding Approach*, New York, Oxford University Press.

Pylyshyn, Z. W. (1984). *Computation and Cognition: Toward a Foundation for Cognitive Science*; MIT Press, Cambridge MA.

Reeker, L., T.A. Blaxton, & C.R. Westphal

Rosche, E., C.B. Mervis, W.D. Gray, D.M. Johnson, & P. Boyes-Braem (1975). Basic objects in natural categories. *Cognitive Psychology, 8,* 382–439.

Schank, R. C. and R. Abelson (1977) *Scripts, Plans, Goals, and Understanding.* Hillsdale, NJ: Lawrence Earlbaum, Associates.

Vekker, L.M. (1981) *Mental Processes: The Agent, the Experience, Action and Consciousness* Vol. 3, Leningrad University Press. (In Russian).

Vekker, L.M. (1976) *Mental Processes: Thinking and the Intellect.* Vol. 2, Leningrad University Press. (In Russian).

Vekker, L.M. (1974) *Mental Processes.* Vol. 1, Leningrad University Press. (In Russian).

Westphal, C.R. (1988). Using simulation-like methodologies for knowledge base validation. *Proceedings of the Third International Symposium on Knowledge Engineering.*

APPENDIX A: LIST OF PUBLICATIONS FROM BDM-KAT PROJECT

Blaxton, T.A. & L.H. Reeker A tool for knowledge acquisition in process domains

Blaxton, T.A. & C.R. Westphal (1988) Combining explicit queries with simulation techniques during knowledge acquisition. *Proceedings of the 1988 Eastern Simulation Conference.*

Kushner, B.G., R.A. Geesey, P.A. Parrish, & S.G. Wax (1987). A knowledge acquisition tool for the intelligent processing of materials. In R.J. Harrison & L.D. Roth (eds) *Artificial Intelligence Appliciations in Materials Science.* The Metallurgical Society, Inc.:Warrendale, PA.

Lancaster, J.S. (1990). Cognitively-based knowledge acquisition: An integrated toolkit. To appear in McGraw, K.L. & C.R. Westphal (eds) *Readings in Knowledge Acquisition: Current Practices and Trends.*

Lancaster, J.S., Vekker, L.M., & Kushner, B.G. (1989). Rapid prototyping of intelligent controller knowledge bases u;sing the BDM knowledge acquisition toolkit. Presented at the Annual Meeting of the Minerals, Metals, and Materials Society.

Lancaster, J.S., C.R. Westphal, & K.L. McGraw (1989) A cognitively valid knowledge acquisition tool. *Special SIGART Issue on Knowledge Acquisition*, Association of Computing Machinery.

Levy, A.B. & J.J. Wlassich (1988). Knowledge-based systems applied to advanced materials manufacturing.

McGraw, K.L., C. Westphal, & J.S. Lancaster (1989). Applying hypertext techniology to integrate multiple reasoning models in BDM-KAT. *Proceedings of the Sixth Annual Workshop on C2 Decision Making*, Joint Directors of Laboratories, San Diego, CA.

Reeker, L.H., T.A. Blaxton, & R.A. Geesey (1987) A knowledge acquisition tool for processes. Presented at the Fifth Intelligence Community AI Symposium, March 1987.

Reeker, L.H., T.A. Blaxton, & C.R. Westphal (1988) Knowledge engineering as software engineering--and vice-versa. Presented at the 27th Annual Technical Symposium, Washington, D.C. Chapter of the Association of Computing Machinery.

Westphal, C.R. (1988) Knowledge engineering for software design. *ACM Software Engineering Notes.*

Westphal, C.R. (1988) Using simulation–like methodologies for knowledge base validation. *Proceedings of the Third International Symposium on Knowledge Engineering.*

A TOOL FOR KNOWLEDGE ACQUISITION IN PROCESS DOMAINS

Extended Abstract

T. A. Blaxton and L. H. Reeker
The BDM Corporation, Advanced Technology Group
7915 Jones Branch Drive, McLean, VA 22102

1. Introduction: Knowledge Engineering and KAT

Today's applications of artificial intelligence are based upon the fact that useful tasks can be accomplished by programs containing large, detailed, specialized knowledge bases. It has become apparent that one of the major obstacles in applying artificial intelligence to specialized domains is the difficulty of creating these knowledge bases, which requires the acquisition of the requisite knowledge from human experts. The Knowledge Acquisition Tool (KAT) under development at The BDM Corporation is designed to help overcome this obstacle in the area of knowledge acquisition about processes or procedures.

In contrast to using a tool specifically designed for knowledge acquisition, the typical approach to building knowledge bases entails prolonged and intensive one-on-one interactions between a programmer and a domain expert. In this setting the programmer may elicit large amounts of information from the expert, impose his/her own organization onto this information, and encode it into the system. An obvious drawback to this approach is that the programmer ends up learning much more detail about the domain than is likely necessary, and the expert becomes far more knowledgeable about AI programming than he or she might like. Although there have been many recent advances in expert system building technology such as expert system shells, these advances do not address the knowledge acquisition bottleneck.

Figure 1 shows the typical life cycle of a knowledge-based system, based on that for a conventional software system (Jackson, 1975). The dotted portion is what is usually called "knowledge engineering", consisting of knowledge acquisition and the initial design and prototype development, based on the knowledge acquired (though in the latter, the knowledge engineer may be aided by other programmers). For a given process domain, KAT is intended to replace this portion. We wish to state from the outset that we have not tried to design KAT as something that will evolve into an expert system for materials processing, or even as a tool for building an expert system directly. Rather, KAT will be used by the programmer to build up a knowledge base that can then be handed over to designers who will be building an expert system. The knowledge base will not provide all of the interface features, the real-time performance, or even the inferencing capabilities that are desired in the final system. Although KAT does contain within it a certain amount of "expertise" in the methodology of process design, the structure is more that of a software design environment than an expert system. It uses an expert system shell on a LISP machine (KEE/SIMKIT on the Symbolics 3670), but the purpose of using these systems is to

Work described in this paper was supported in part by the U. S. Defense Advanced Research Projects Agency, under contract number F49620-86-C-0036.

provide appropriate structures for building the knowledge base and stepping through the dynamics of the process.

Although it will automate the knowledge acquisition process to some degree, then, KAT is not envisioned as a replacement for the knowledge engineer. It is, in fact, being designed with the objective in mind that it will be used by a knowledge engineer, working with an expert or multiple experts. At present, KAT is expected merely to alleviate the usual situation that requires the knowledge engineer to "become an expert" on the domain in question, because KAT can be used to record and refine the expert's knowledge in an incremental fashion.

Using KAT, the knowledge engineer and domain expert will iterate through three querying modes in which the expert will be prompted to answer different types of questions about a particular materials processing domain (see Figure 2). In the Clarification Mode, the expert will be provided with facilities for creating an AND/OR graphical representation of the process. Once a certain amount of information is encoded in this format, the system will switch into a Prediction Mode. At the start of the Prediction Mode, the knowledge structure created during the Clarification Mode will be converted into a flowchart display. The user will be prompted to answer a series of questions about each subprocess represented in that structure. For instance, the expert might be asked to estimate ranges for temperature, pressure, and exhaust gas ratios at each point identified so far for the process. In addition, the expert will be asked to make predictions about what could go wrong at each given point, and to provide suggestions for redesign that might lead to avoidance of identified problems. Once these queries are completed, the knowledge engineer will return to the Clarification Mode, and recycle through the questioning, adding to the knowledge base created thus far.

Once the expert has iterated through the first two modes several times, a fairly detailed knowledge structure will be in place. At this point, the Diagnosis Mode will be invoked. In this mode, KAT's facilities will provide a step-through presentation of the materials process as specified up to this point. Sensor readings will be displayed so that the user can see how they change dynamically as the process progresses. In addition, the user will be able to trace the pattern of rules that are being evoked to affect these changes along the way.In this environment, then, the expert will have opportunities to diagnose the knowledge base created thus far, adding and deleting elements as necessary.

As mentioned above, KAT is being built at BDM on a Symbolics 3670 in the KEE and SIMKIT environments. KEE is used because it provides many facilities to aid AI programmers in the generation of frame-like objects and rules. In addition, it has graphics and specialized mouse functions which are useful in constructing user interfaces. The SIMKIT system, which resides on top of KEE, provides a host of facilities for writing event-driven simulations including a clock, event calendar, data collectors, and various mathematical tools for data analysis. The event-driven nature of the SIMKIT simulation environment makes it ideal for the materials processing domain.

KAT is a sophisticated tool, embodying the best methodologies known for process specification, both in top-down and a bottom-up fashion. The top-down portion is based on the highly-successful Higher Order Software approach (Hamilton and Zeldin, 1980; Harel (1980)) used in complex software development for NASA, extended

slightly to allow parallel processes (Reeker, 1986). The bottom-up portion features flowcharting that can be rearranged dynamically and transformed back to the top-down AND/OR representation.

KAT extracts knowledge both through explicit and implicit queries of the expert, who is being aided in the use of KAT by the knowledge engineer. It has been suggested that procedural knowledge might best be extracted by implicit means (see, e.g., Graf and Schacter, 1985). In specifying processes however, all knowledge is not procedural, as a number of declarative facts will be germane to the process [for the distinction between procedural and declarative knowledge, see Kolers and Roediger (1984)]. The three modes of query shown in Figure 2 are designed to elicit different types of knowledge in specialized ways. A more detailed description of knowledge acquisition in each of these modes will now be provided, along with a discussion of other features of KAT.

1.1. The Materials Processing Library

Before one can build a process model in KAT, a library must be constructed which contains a sampling of the objects and rule classes that will be used in the final system. Our initial implementation of KAT is in the domain of materials processing, specifically chemical vapor infiltration (CVI). For this domain, objects in the library might include reactor types, preform types, information about thermal and pressure gradients, and so on. For each of these objects, further information may be stored in internal "slots". For instance, the object corresponding to a one-dimensional preform might have slots for such features as size, material makeup, initial density, optimum density, and reactive gas, to name a few. Each of these slots is allowed to take on certain values. These values can be set and reset for any particular materials design scenario.

Our aim in constructing the initial library is not to represent exhaustively every element of interest in the materials process. That task will be left to the domain expert when KAT is actually used to build the knowledge base. Rather, the intent is to anticipate, to a certain degree at least, some general concepts that are bound to be needed to describe the CVI process. These basic components will then be in place in the system when the knowledge engineer and the expert sit down for the first time to create a process model. Building this initial library will not only save the expert time, but will provide some guidance for the expert's early efforts in partitioning the CVI world into objects.

1.2. The Clarification Mode

The clarification mode (Figure 3) may be thought of as a framework that allows the expert to tell the system what each process consists of, in terms of immediate subprocesses, as the expert sees them. This is the top-down, or process decomposition, portion. As mentioned above, the result is an AND/OR graph. Where OR nodes are used, the branches will often be labeled by conditions that would cause one to choose one of the alternatives, by question marks to indicate that the alternatives have not yet been decided, or by the initials of one or more of multiple experts.

At the outset of the clarification mode, certain objects corresponding to units in the materials library will be available in an object window. If the expert wishes, new objects not already in the library can be created at any time and placed in this object window. The expert will indicate that a particular object should be moved into the

main viewport and included in the AND/OR graph. Once several objects are in place in the viewport, the expert can then indicate the type of relation that should exist between them. These relations include AND, OR, and NEXT, and will be displayed in the relations window next to the viewport. The final configuration will be displayed with arrows indicating relations that have been defined between objects. This graph creation will continue until the expert has thoroughly specified the process.

1.3. Prediction Mode

The prediction mode (Figure 4) provides another view of the same process, in terms of process flow, exposing "holes" that were missed in the clarification mode and allowing them to be filled either in prediction mode (i.e. bottom-up) or by return to clarification mode. Time and sequencing relationships that may only be implicit in the AND/OR representation are now shown explicitly, though the logical subsetting process explicit in the AND/OR graph is suppressed.

The information entered into the system during the Clarification mode is converted into a flowchart representation before being shown again in the Prediction mode. In order to do this, AND and OR tree structures are collapsed down into single composite objects. These composite objects are displayed with simple "next" relations in the flowchart format, thus emphasizing the temporal flow of the process. These composite objects may be expanded into their parts as desired.

For each subprocess point in the flowchart, the expert is asked which sensors are relevant and what their normal ranges should be. In addition, the expert is asked questions concerning his/her interpretation of sensor readings at each subprocess point. The responses to such queries take the form of rules. For example, the expert might say something like "If the temperature of the reactive chamber is greater than 1200 degrees, the infiltration of the preform will be uneven". Menu-driven user interface facilities handle the construction of rules in cases such as this. It is likely that this identification of potential problems will prompt the expert to redesign the process in some cases. Thus facilities for rearranging, adding, deleting, and creating new objects are also available during the Prediction mode.

Once the expert has been queried about each subprocess in the flowchart, the knowledge engineer returns to the Clarification mode. Here, the knowledge structure is presented in an AND/OR format and the expert will further specify the process. It is expected that switching between the AND/OR graph and the flowchart displays will prompt the expert to think of details to add to the model created up to that point. These details might not otherwise be remembered were the model always presented in one format.

1.4. Diagnosis Mode

By the time the expert moves into diagnosis, the clarification and prediction modes will have been iterated many times. The diagnosis mode (Figure 5) provides a means of "stepping through" the acquired knowledge in a dynamic manner, indicating consequences of the process in whatever detail has been given at any time. The detail might initially be quite sketchy, but as more and more knowledge is added, it becomes more explicit. Where it is not adequately explicit, the diagnosis mode may expose a need for more detail. In a sufficiently knowledge-rich environment, the diagnosis mode could actually *simulate* the process taking place. For the materials processing case, we have looked at the knowledge required, and it is considerable, including

qualitative physical knowledge of the sort studied by De Kleer (1984), and Kuipers (1984). Though we have interest in this aspect of KAT for the future, the diagnosis mode now consists only of an event-stepped "tour" of the process specified, in which the expert will have to observe to see that events are taking place as has been specified, and that conditions specified for the events are, indeed, true.

As may be seen in Figure 5, specialized graphics will be created for displaying sensor readouts. In addition, the expert will be able to observe directly the consequences of process rules defined during the Prediction mode. If at any time the expert perceives the process to be going awry, s/he can stop it immediately and redesign as necessary. Such redesign could include any number of measures such as adding, deleting, rearranging sensors, rules, or objects.

Note that the form of query in the Diagnosis mode is more implicit than in the previous modes. There is no program of queries to which particular types of responses are expected. That is, the expert is not being given direct questions about the structure of the process. Rather, s/he is allowed to observe a step-through "simulation" as defined by information provided thus far, and then make determinations about how this design could be improved. This progression through previously defined states is displayed through, among other things, changes in sensor icons.

2. The general import of KAT

The basic KAT system is very general as to the type of process or procedure being discussed. It allows both parallel and serial processes, and allows specification of alternative processes where there is uncertainty or disagreement as to steps to be taken. It can thus be used by a single expert or multiple experts. In extracting information from multiple experts, KAT will provide an opportunity that may be unavailable in the conventional domain. Multiple experts may be geographically distributed or have other reasons that they cannot come together to give information. But the knowledge engineer can obtain the information from them separately, then just focus on the cases where KAT indicates that they have disagreed.

A comparison of KAT to the traditional knowledge engineering approach is given in Figure 6. In general, KAT will relieve the knowledge engineer from the necessity of "becoming an expert" in the subject; will facilitate incremental knowledge acquisition, thus making the most of the expert's limited time; will provide a facility for allowing aspects of the process to be represented and understood more fully; and will provide an archival form for the body of knowledge, which will help in the modification or updating of the expert system and in the design of related systems.

It is clear that knowledge engineering will be important in the future not only for the development of knowledge-based systems but as a form of archiving expertise in a condensed form for multiple purposes. These include reference use and education, as well as the building of various forms of systems, designing processes, etc. We feel, therefore, that knowledge engineering tools will have wide applicability. But the immediate uses for which KAT is designed are in the expert system area. Some potential areas of expert system development which we are exploring with an eye to the use of KAT are shown in Figure 7.

References

Bobrow, D. (1985), *Qualitative Reasoning about Physical Systems*, MIT Press, Cambridge Mass.

De Kleer, J. (1984). How circuits work, *Artificial Intelligence*, **24**. Reprinted in Bobrow (1984), 7-831

Graf, P., and D. Schacter (1985). Implicit and explicit memory for new associations in normal and amnesic subjects. *Journal of Experimental Psychology: Learning, Memory and Cognition*, **11**, 501-518.

Hamilton, M., and S. Zeldin (1976). Higher-order software — a methodology for defining software, *IEEE Trans. Software Engineering*, **SE-2**, 1, 9-32.

Harel, David (1980). AND/OR programs: A new approach to structured programming, *ACM Transactions in Programming Languages and Systems*, **2**, 1, 1-17.

Jackson, M. A. (1975). *Principles of Program Design*, Academic Press, New York.

Kolers, P. A., and H. L. Roediger (1984). Procedures of mind, *Journal of Verbal Learning and Verbal Behavior*, **23**, 425-449.

Kuipers. B. (1984). Commonsense reasoning about causality: Deriving behavior from structure, *Artificial Intelligence*, **24**. Reprinted in (Bobrow, 1985), 169-204.

Reeker, L. H. (1986). Extended AND/OR graphs and parallel programming in a graphic environment, *Proceedings of International Computing Symposium 1986*, Tainan, Taiwan, December 1986, 138-146.

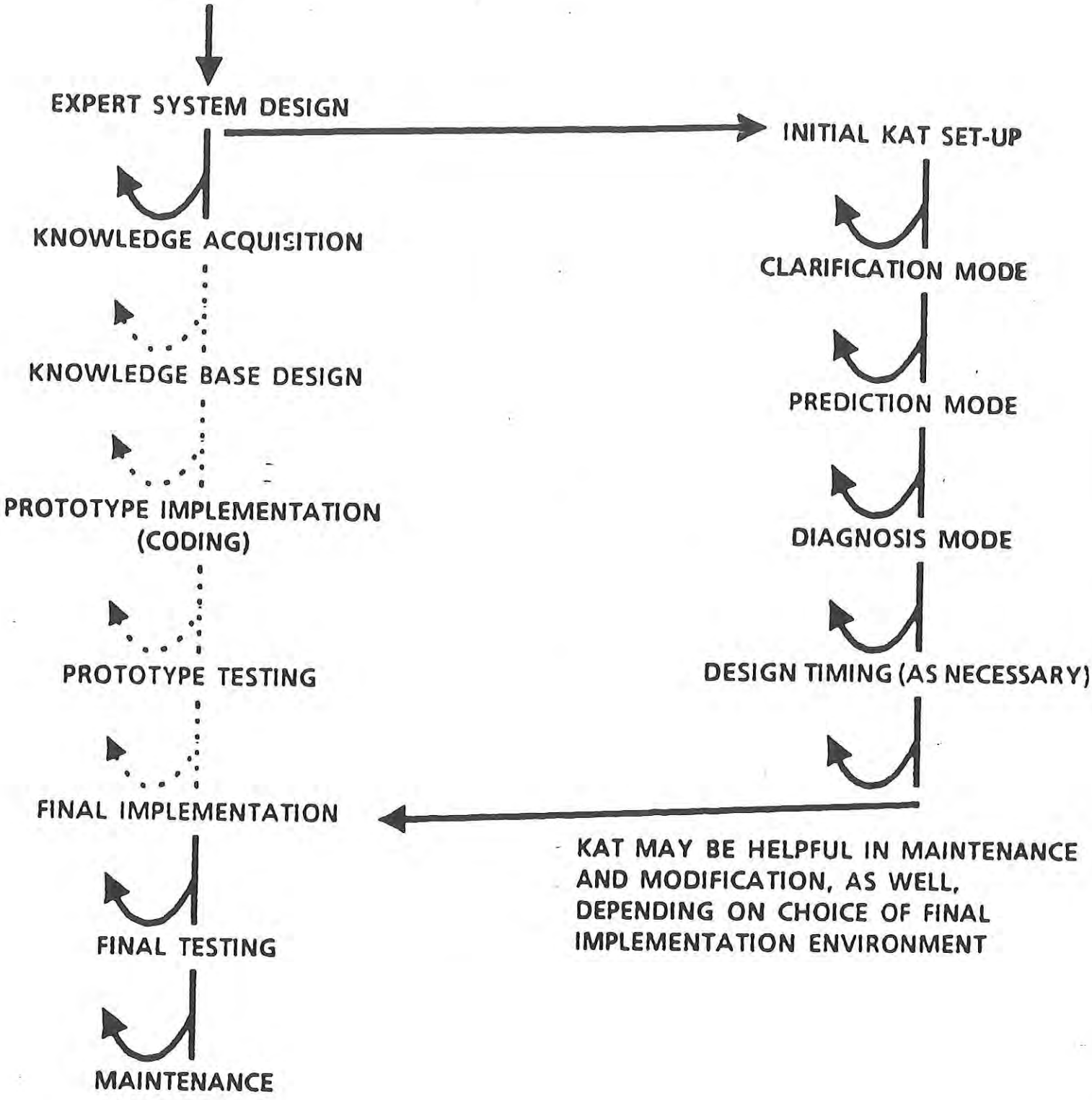
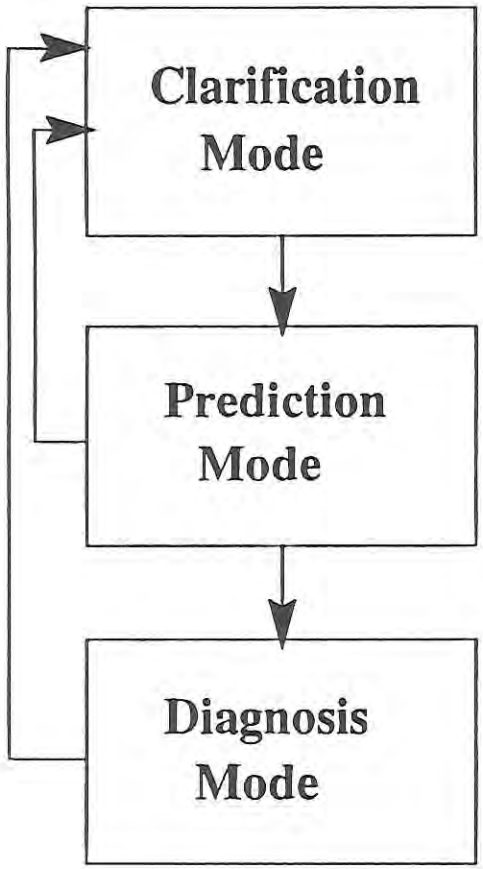


Figure 1. KAT IN THE EXPERT SYSTEM LIFE CYCLE

THE OVERALL STRUCTURE OF KAT



+ Description of process decomposition is created in the form of an AND/OR graph

+ Logical relationships among processes are represented explicitly

+ Information is presented in flowchart format, and expert checks -- inputs & outputs
-- relevant attributes
-- process rules

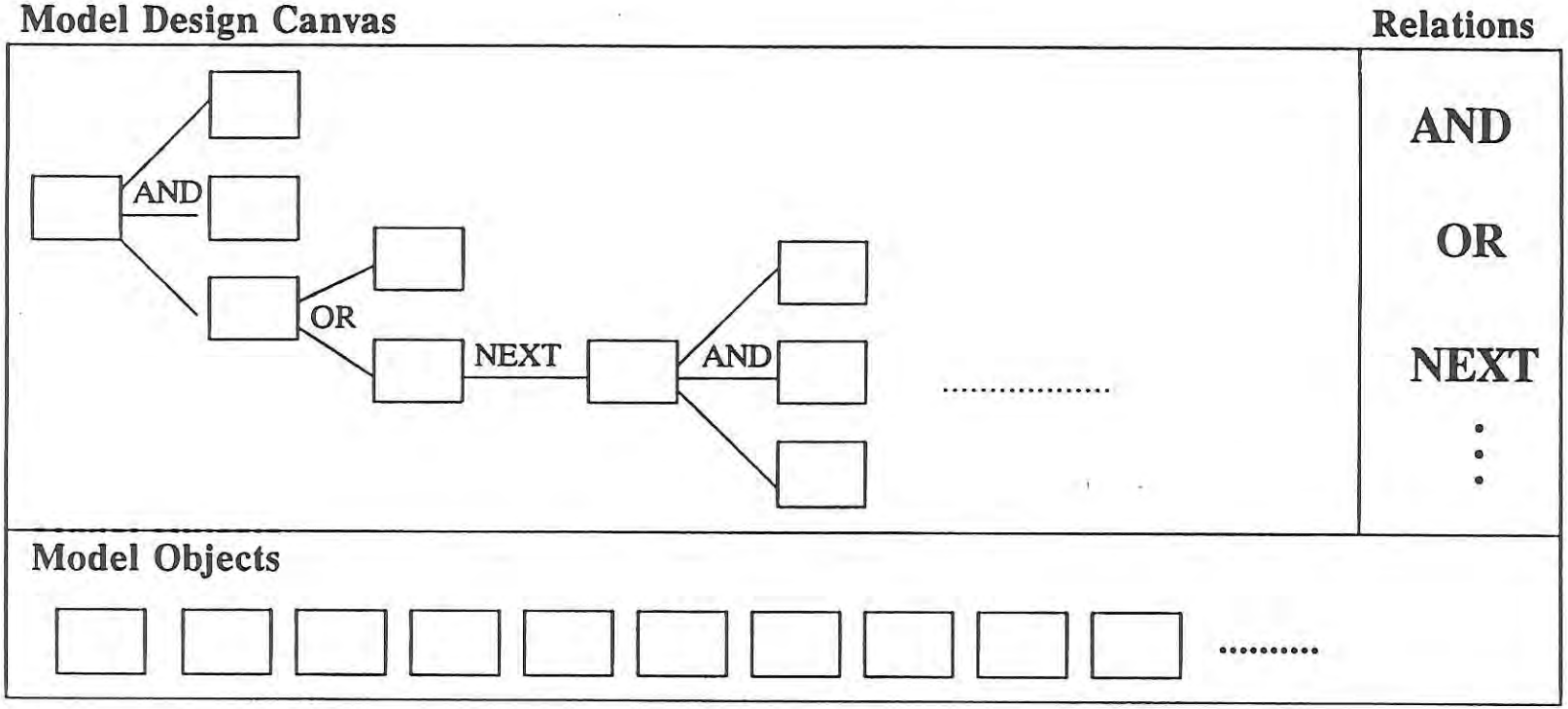
+ Information format is changed to show temporal ordering; procedural knowledge elicited.

+ Dynamically step through knowledge structure, making changes as necessary

+ Implicit interactions within the process model are explicated

Figure 2

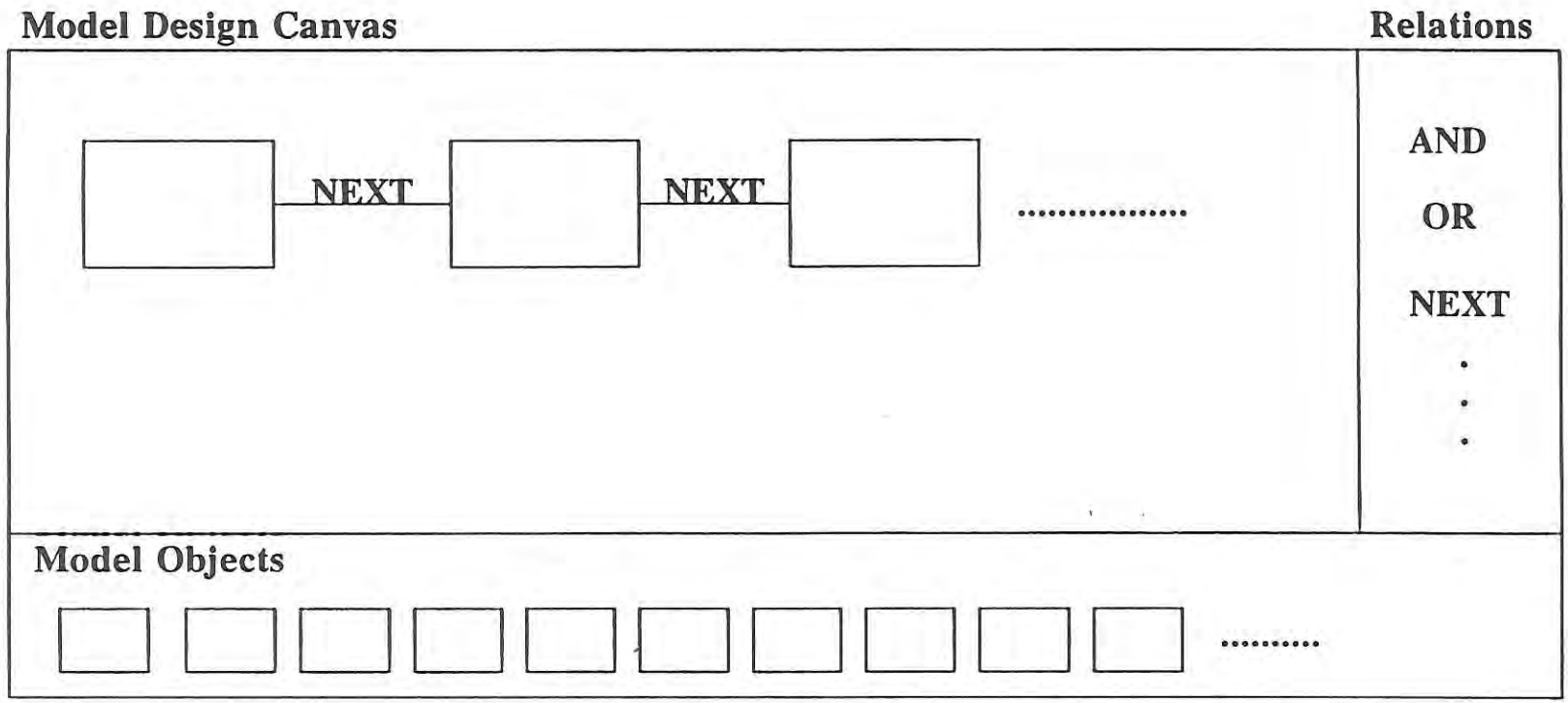
CLARIFICATION MODE



+ Expert starts from scratch and decomposes process into a logical progression of subprocesses.
+ Alternative subprocesses are indicated where appropriate (ORs).
+ Result is an AND/OR graph of the entire process

Figure 3

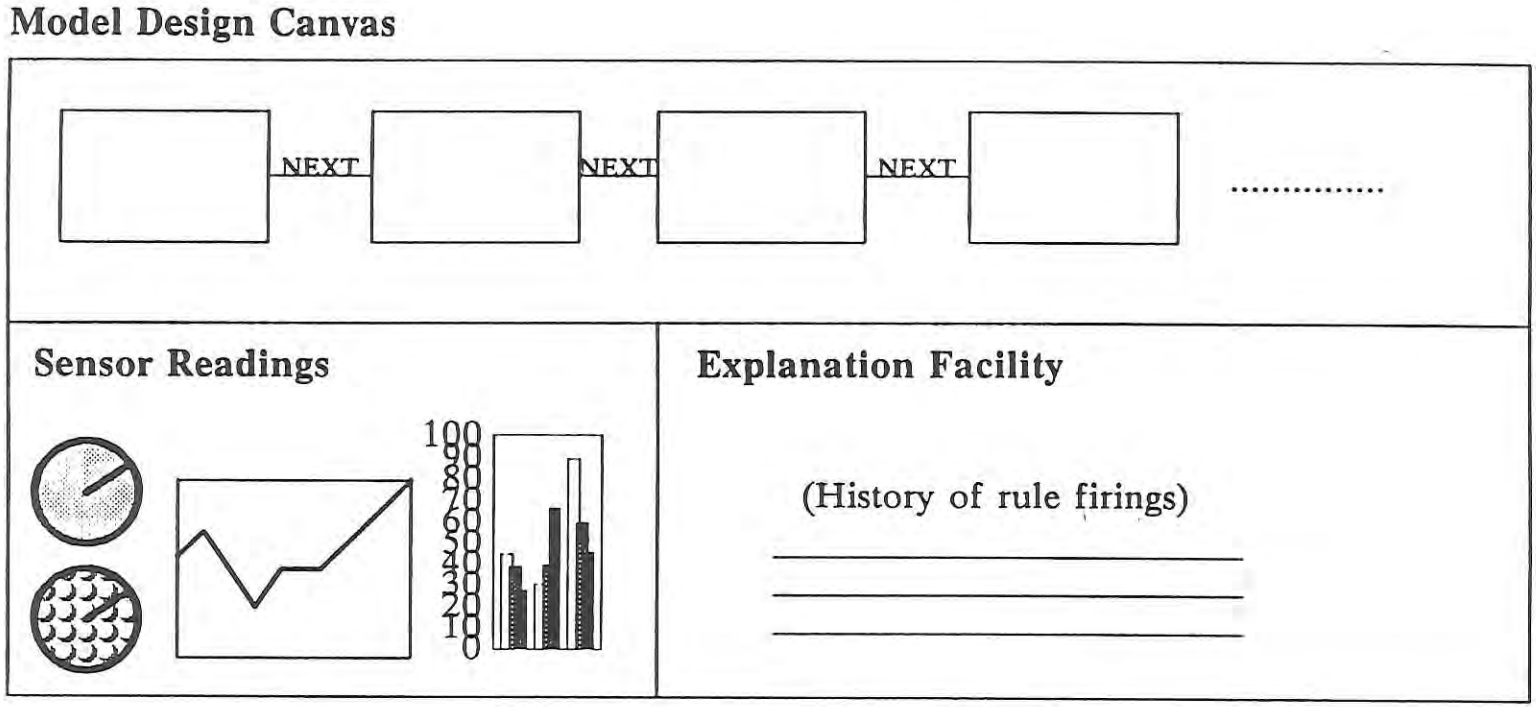
PREDICTION MODE



+ Knowledge structure created in the Clarification Mode is redisplayed in a temporal (flowchart) form.
+ For each subprocess, the expert identifies:
  - Inputs and outputs
  - Other relevant data structures
  - Rules which define relationships among subprocesses, etc.
+ Temporal ordering may be easily modified.

Figure 4

DIAGNOSIS MODE



+ Knowledge structure is presented in flowchart form.
+ Expert sees a dynamic step–through display of the process with relevant sensor readings and the history of rule applications.
+ This step–through serves as an implicit query, prompting the expert to redesign upon noticing incomplete or faulty specifications.

Figure 5

COMPARISON OF KAT WITH TRADITIONAL APPROACHES TO KNOWLEDGE ENGINEERING

TRADITIONAL KNOWLEDGE ENGINEERING	KAT
LIMITED (EXPLICIT) TECHNIQUES FOR ELICITING KNOWLEDGE	LARGE NUMBER OF TECHNIQUES FOR ELICITING KNOWLEDGE (IMPLICIT AND EXPLICIT)
INFORMATION RECODING: - TIME CONSUMING - LOSS OF INFORMATION WHEN SWITCHING FORMATS - KNOWLEDGE MAY END UP IN AN INAPPROPRIATE FORM	- NO RECODING OF INFORMATION; ALL INPUTS ARE PUT DIRECTLY INTO THE KB - SYSTEM CONTROLS THE TYPE OF DATA STRUCTURE USED TO REPRESENT INFORMATION
INTEGRATION OF INPUTS FROM MULTIPLE EXPERTS IS DIFFICULT	SPECIALLY DESIGNED FOR USE WITH MULTIPLE DOMAIN EXPERTS
PROCESS FOR VALIDATING THE KB IS ILL-DEFINED	DIAGNOSIS MODE OF QUERY IS DESIGNED TO ASSIST IN THE VERIFICATION PROCESS

Figure 6

POTENTIAL AREAS FOR FURTHER APPLICATIONS

++ In principle, KAT is applicable to a variety of expert system building domains. Some of these might include:

-- Materials Selection
-- Factory Automation
-- Diagnosis/ Maintenance
-- Planning Courses of Action
-- Crisis Management
-- Concealment, Cover, and Deception
-- Software Engineering

Figure 7

AI PAPERS, 1988

Proceedings of the Conference
on AI and Simulation
18-21 April 1988
Orlando, Florida

Edited by
Ranjeet J. Uttamsingh, PhD
Synetics

Simulation Series
Volume 20
Number 1
January 1989

A Society for Computer Simulation International (Simulation Councils, Inc.) publication.
San Diego, California

AI Papers, 1988
© by The Society for Computer Simulation International
ISBN 0-911801-35-9

Combining explicit queries with simulation techniques during knowledge acquisition

Teresa A. Blaxton
and
Christopher R. Westphal

The BDM Corporation
McLean, VA

ABSTRACT

Despite recent advances that have been made in expert system building technologies, the process of obtaining knowledge from human experts and coding it into a computing system remains fraught with difficulties. The Knowledge Acquisition Tool (KAT) currently being developed at the BDM Corporation is designed to address this knowledge acquisition bottleneck in the domain of materials processing. KAT offers the knowledge engineer a variety of modes in which to query a materials expert on different aspects of a given materials domain. In the Clarification Mode, the expert specifies the entire process in terms of an AND/OR graph. This structure is converted to a flowchart in the Prediction Mode, where the expert is asked to predict problems that can occur at different points in the process, and to specify limiting conditions. Finally, the Diagnosis Mode provides a dynamic step-through of the process model which is somewhat analogous to a simulation. This step-through provides an opportunity for the expert to verify the specification of the process and redesign as necessary.

INTRODUCTION AND OVERVIEW

Today's applications of artificial intelligence are based upon the fact that useful tasks can be accomplished by programs containing large, detailed, specialized knowledge bases. It has become apparent that one of the major obstacles in applying artificial intelligence to specialized domains is the difficulty of creating these knowledge bases, which require the acquisition of the requisite knowledge from human experts. This knowledge acquisition bottleneck reflects the difficulty of obtaining knowledge from domain experts in a structured fashion and converting it into a form that can be readily accessed by the expert system itself. There have been many recent advances in expert system building technology such as expert system shells, but these advances do not address the knowledge acquisition bottleneck. Building the initial knowledge base remains one of the largest stumbling blocks to successful system completion.

The typical approach to building knowledge bases for AI systems entails prolonged and intensive one-on-one interactions between a programmer and a domain expert. In this setting the programmer may elicit large amounts of information from the expert, impose his/her own organization onto this information, and encode it into the system. An obvious drawback to this approach is that the programmer ends up learning much more detail about the domain than is likely necessary, and the expert becomes far more knowledgeable about AI programming than s/he might like.

The Knowledge Acquisition Tool (KAT) project, currently underway at BDM, is aimed at investigating remedies for the knowledge engineering problem. KAT is being designed for use by a programmer as a tool to aid in the knowledge acquisition process, providing structured mechanisms for eliciting information about materials processing. We wish to state from the outset that KAT will not be used as a tool for building an expert system in materials processing. Rather, KAT will be used by the programmer to build up a knowledge base that can then be handed over to designers who will be building an expert system. Furthermore, it is important to realize that although KAT will automate the knowledge engineering process to some degree, we do not envision it as a replacement for the knowledge engineer. Instead it is best thought of as a tool that the knowledge engineer uses to query the expert in a structured manner.

KAT can be used to extract knowledge both through explicit and implicit queries of the expert, who will be aided by the knowledge engineer. Research in the area of human memory has suggested that procedural knowledge might best be extracted by implicit means (e.g., Graf & Schacter, 1985; Roediger & Blaxton, 1987). In specifying processes, however, all knowledge is not procedural, as a number of declarative facts will be germane to the process. (For a discussion of the distinction between procedural and declarative knowledge, see Kolers & Roediger, 1984.) For this reason, KAT has been designed to provide facilities for obtaining both types of knowledge in a natural manner.

As shown in Figure 1, the expert will interact with KAT in three different "modes" called Clarification, Prediction, and Diagnosis. These modes will be used iteratively to provide a structure for knowledge acquisition. The Clarification Mode allows the expert to define the overall process in terms of an AND/OR graph of subprocesses. That is, the expert tells the system what his/her mental model of the process is in terms of subprocesses. The Prediction Mode will provide another view of the model built up during the Clarification Mode. The already specified knowledge structure will be presented in such a way as to emphasize the temporal flow of the process, thus exposing "holes" that were missed earlier. Time and sequencing relationships that may only be implicit in the AND/OR representation are now shown explicitly, though the logical subsetting apparent in the AND/OR graph is suppressed. During this mode the user will be asked questions concerning particular conditions that must be satisfied in order for the process to continue and so on.

Once the expert has iterated between the Clarification and Prediction Modes several times, a fairly well-specified knowledge structure will be in place and the Diagnosis Mode will be invoked. The Diagnosis Mode will provide a means of "stepping through" the process model in a dynamic manner.

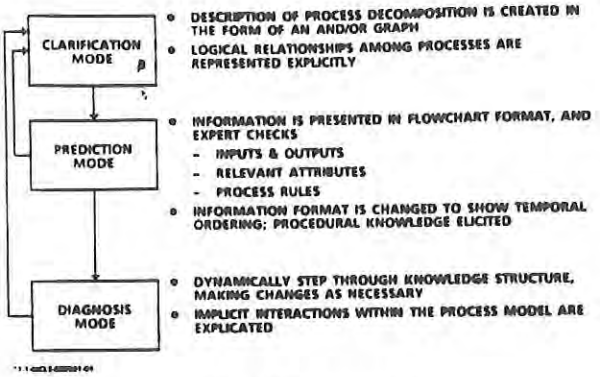


Figure 1. The Overall Structure of KAT

Where the model is not yet adequately specified, this mode will expose the need for more detail. The step-through will resemble a simulation on the surface, although it will not really be a simulation in the truest sense of the word. In a sufficiently knowledge-rich environment, the Diagnosis Mode could actually simulate the process taking place. For a materials processing application, the amount of knowledge required for a simulation would be considerable and would include the sort of qualitative physical knowledge studied by De Kleer (1984) and Kuipers (1984). Though we have interest in this aspect of KAT for the future, the present Diagnosis Mode is designed to do an event-stepped "tour" of the process specified, in which the expert will have to verify that the model is indeed reflective of the true physical world.

KAT VERSUS TRADITIONAL KNOWLEDGE ENGINEERING TECHNIQUES

As was already mentioned, the traditional approach to knowledge engineering usually involves an extended interviewing process whereby the knowledge engineer asks the expert as many questions as possible about the domain of interest. As shown in Figure 2, these questions are limited in their utility in several ways. First, the queries are explicitly formed and are presented to the expert in such a manner as to presuppose a particular type of response. Second, the fact that the queries can be posed at all by the knowledge engineer implies that s/he has some understanding of the domain and what its important information elements are. In all likelihood a large amount of time will be spent in coming up to the level of understanding that enables meaningful interactions with the expert.

The third and most serious drawback to the questions that the knowledge engineer poses to the expert is that they reflect the knowledge engineer's own mental model of the domain and not the expert's. The knowledge engineer deems particular questions to be important because they fill in missing information in his/her personal conception of the problem domain. Since the knowledge engineer is not experienced in the domain, it is quite likely that s/he will key in on concepts that are only tangential to the main issues, and thus will bias the construction of the knowledge base. Since the expert is not experienced in AI programming, s/he will not be able to look at the knowledge structure as it is developed and make corrections as to the proper hierarchical arrangement of knowledge base elements. One might imagine how a few mistakes of this type early on in the knowledge acquisition process could lead to deeply entrenched biases in the knowledge structure that would be difficult to compensate for later.

TRADITIONAL KNOWLEDGE ENGINEERING	KAT
LIMITED (EXPLICIT) TECHNIQUES FOR ELICITING KNOWLEDGE	LARGE NUMBER OF TECHNIQUES FOR ELICITING KNOWLEDGE (IMPLICIT AND EXPLICIT)
INFORMATION RECODING: - TIME CONSUMING - LOSS OF INFORMATION WHEN SWITCHING FORMATS - KNOWLEDGE MAY END UP IN AN INAPPROPRIATE FORM	- NO RECODING OF INFORMATION: ALL INPUTS ARE PUT DIRECTLY INTO THE KB - SYSTEM CONTROLS THE TYPE OF DATA STRUCTURE USED TO REPRESENT INFORMATION
INTEGRATION OF INPUTS FROM MULTIPLE EXPERTS IS DIFFICULT	SPECIALLY DESIGNED FOR USE WITH MULTIPLE DOMAIN EXPERTS
PROCESS FOR VALIDATING THE KB IS ILL-DEFINED	DIAGNOSIS MODE OF QUERY IS DESIGNED TO ASSIST IN THE VERIFICATION PROCESS

Figure 2 Comparison of KAT With Traditional Approaches to Knowledge Engineering

In contrast to this approach, KAT offers several modes in which the expert may be queried about the domain. Some of these queries will be explicit whereas others will be implicit. For example, in the Prediction Mode the expert will view a flowchart of the process and will be asked explicitly to identify sensors that are of interest at a particular point in the process. In the Diagnosis Mode, however, a dynamic step-through of the process will be presented. Readings from sensors will be made available to the expert and s/he will infer from these readings whether the process is properly specified. This second type of interaction involves what may be termed an implicit query in that the expert is watching the process in action and redesigning as necessary, rather than responding to overt questions from the knowledge engineer. This is a very different type of methodology than has been employed in other knowledge acquisition systems.

RECODING PROBLEMS

A second class of problems that one encounters when using traditional approaches to knowledge engineering involves the recoding of information provided in the expert's responses to a form that the AI system can use. In most cases, the expert is interviewed at length at some place other than the software development site. Following an interview session, the knowledge engineer has to work through and organize the information gleaned during the interview and then decide how best to encode it into the already existing knowledge structure. Since the expert is not present when this recoding takes place, the knowledge engineer runs the risk of either omitting important information, coding the information into an inappropriate form, or both.

KAT is designed to avoid these recoding problems. Specifically, since the knowledge engineer will be using KAT while knowledge acquisition occurs, the information gathered from the expert may be directly encoded into the knowledge base as it is elicited. Furthermore, the queries posed by the system will be designed in such a way that their answers will naturally fit into a particular data structure format. That is, the system will already be expecting that a response to a particular question will take the form of a rule (or a frame, as the case may be). Facilities for

entering the information into the system will be available to automate this process as much as possible. Thus, errors of the type discussed earlier may be more easily avoided. Finally, since the expert will be present when the information is entered into the knowledge base, s/he will have the opportunity to suggest corrections as appropriate.

MULTIPLE EXPERTS

A classic problem encountered during knowledge engineering efforts is the acquisition of information from multiple experts. In many cases, the experts fail to agree on important points and the knowledge engineer is in no position to judge the opinions one against the other. Perhaps more seriously, the mental model of the domain can vary considerably from expert to expert. This is reflected in differences in emphasis that are imparted to the knowledge engineer during the acquisition process. The fact that one concept is considered more important than another can have a large effect on the knowledge base structure. Consequently, knowledge bases engineered from multiple experts can "look" different from one another, even though there may be few outright contradictions.

KAT is being designed to address this problem. By providing facilities for experts to specify and edit models of the materials process, we will have a handy tool for observing differences in emphasis imparted by different experts. There will be opportunities for an expert to compare directly his/her formulation of the domain with that specified by other experts. Once laid bare, these disagreements can either be resolved by revising one or more of the knowledge structures, or at least can be marked for future reference.

KNOWLEDGE BASE VERIFICATION

One final but important class of problems that KAT addresses is that of verifying the knowledge structure once it is in place. In normal practice, it is sometimes quite difficult to check that the information encoded in the system actually conveys the meaning intended by the expert. Although there exist no standard set of procedures for establishing the credibility of a system, many environments perform at least some form of static overview. However, these checks provide only limited insurance that the model is correct, and unless a system is validated its credibility will suffer. A more promising approach is the use of non-static evaluation techniques such a simulation that allow the behavior of a model to be viewed as interactions among its internal objects (e.g., Payne, 1982).

This is analogous to what occurs in the Diagnosis Mode. During Diagnosis, the expert sees an automated "step-through" of the process as it has been specified up to that point. If all is well, the process should flow according to the expert's expectations. Deviations from those expectations, however, indicate either that information has been coded in an improper way or that information is missing altogether. Once these problems are detected, the expert can immediately direct their correction on-line. S/he can then check to see that proposed solutions have indeed eliminated the problem by re-running the step-through.

The Diagnosis Mode uses a discrete-event scheduling method where the rule actions determine the system behavior. The initial specification may be preliminary, thus exposing underdeveloped sections of the model, but as additional knowledge is obtained, it will become increasingly explicit. Since the model uses objects as input/output processes, a model inference (Mihram, 1973) or sensitivity analysis can be applied to the model objects to validate the system performance and obtain variability estimates via multiple process simulations. In addition these causal chains (e.g., Reddy, Fox, & Husain, 1986) may help express alternative parameters for system operations (see Baskaran & Reddy, 1984; Miller, 1974).

INITIAL IMPLEMENTATION OF KAT

KAT is being built at BDM on a Symbolics 3670 machine in the KEE and SIMKIT environments. KEE is an expert system building shell sold by Intellicorp. It provides many facilities to aid AI programmers in the generation of frame-like objects and rules. In addition, it has graphics and specialized mouse functions which are useful in constructing user interfaces. The SIMKIT system, which resides on top of KEE, provides a host of facilities for writing event-driven simulations including a clock, event calendar, data collectors, and various mathematical tools for data analysis. The event-driven nature of the SIMKIT simulation environment makes it ideal for the materials processing domain.

What follows is a description of the features being implemented in KAT using both functions available from the KEE and SIMKIT environments as well as other code generated at BDM. Some of these features have already been implemented, and others are currently being designed.

1. The Clarification Mode

The expert's task in the Clarification Mode is to create an AND/OR graph of the process. At the outset of this mode, certain objects corresponding to units in the materials processing domain are available in an object window (see Figure 3). If the expert wishes, new objects not already present can be created at any time and placed in this object window. The expert indicates that a particular object should be moved into the main viewport and included in the AND/OR graph. When the corresponding icon is moused in the object window, a menu will appear displaying the slots and values of that object before its placement in the viewport is confirmed.

Once several objects are in place in the viewport, the expert can then indicate the type of relation that should exist between them. These relations include AND, OR, and NEXT, and are displayed in the relations window below the viewport. (Note that other relations are displayed as well. These are of interest in the Prediction Mode.) The user mouses on the desired relation, and then mouses on the objects to be linked with that relation. The system then prompts for other objects to be included in the relation, and the final configuration is displayed with arrows indicating relations that have been defined between objects. This graph creation will continue until the expert has thoroughly specified the process.

2. The Prediction Mode

The information entered into the system during the Clarification Mode is converted into a flowchart representation before being shown again in the Prediction Mode (see Figure 4). The AND/OR tree structure is collapsed down into single composite objects. These composite objects are displayed in a

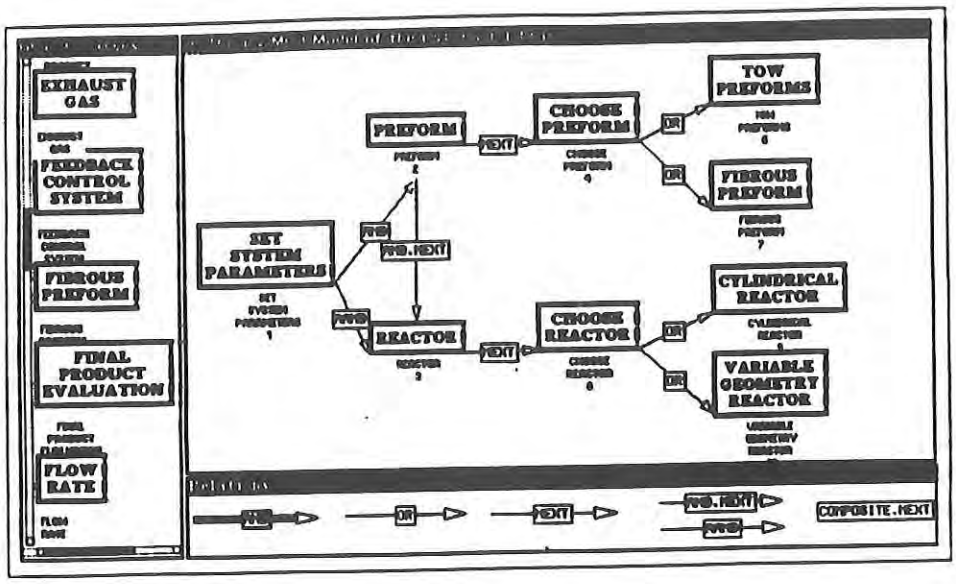


Figure 3. KAT Interface for Clarification Mode

quasi-flowchart format, thus emphasizing the temporal flow of the process.

For each subprocess point in the flowchart, the expert is asked which sensors are relevant and what their normal ranges should be. In addition, the expert is asked what very high or very low readings would indicate at this point in the process. The responses to such queries take the form of rules. For example, the expert might say something like "If the temperature of the reactive chamber is greater than 1200 degrees, then the infiltration of the preform will be uneven". It is likely that this identification of potential problems in the Prediction Mode will prompt the expert to redesign the process in some cases. Thus facilities for rearranging, adding, deleting, and creating new objects are also available during the Prediction Mode.

As one might imagine, the task of handling free-form user inputs and converting them into acceptable rule form is not a trivial one. There are many problems to consider, such as how to have multiple antecedents and consequents, how to handle variables in rules, how to choose from among the several rule types available in the KEE system, and how to incorporate newly created rules into the already-existing knowledge structure. To address these problems, a menu-driven facility is being developed that will handle the construction of rules in a partially automated manner. In its early version, the rule

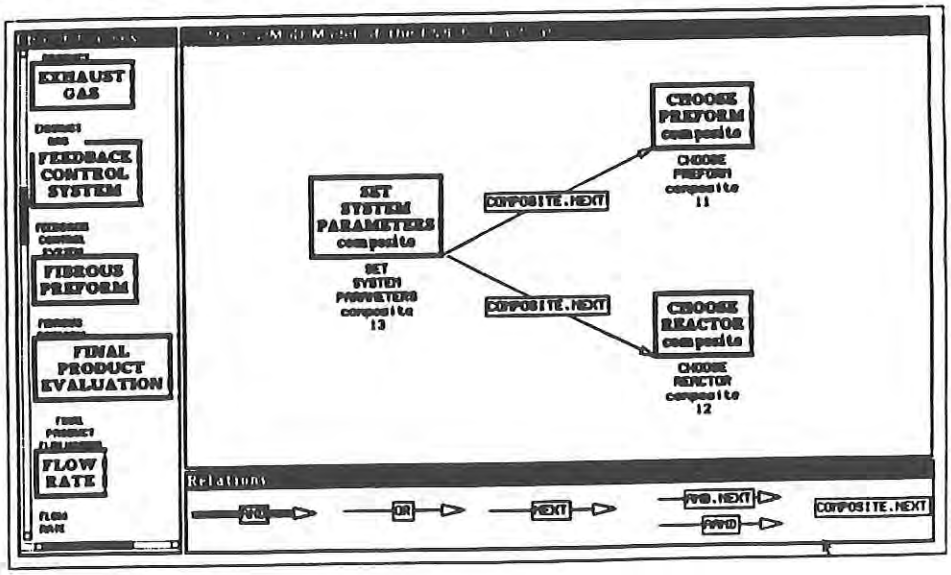


Figure 4. KAT Interface for Prediction Mode

builder will by used by the knowledge engineer to construct standard If-Then rules that conform to the syntax dictated by KEE. Later, facilities will be added for constructing LISP expressions, methods, and so on.

After the expert has been queried about each subprocess in the flowchart, the knowledge engineer returns to the Clarification Mode. Here, the knowledge structure is again presented in an AND/OR format and the expert further specifies the process. It is expected that switching between the AND/OR graph and the flowchart displays will prompt the expert to think of details to add to the model created up to that point. These details might not otherwise be remembered were the model always presented in one format.

3. The Diagnosis Mode

Once the knowledge engineer has iterated through the Clarification and Prediction Modes several times with the expert, a good deal of information about how the process occurs will be represented in the knowledge structure. In the Diagnosis Mode, the expert will have a chance to look at a step-through "simulation" of the process as specified thus far. It is during the Diagnosis Mode session that the features of SIMKIT are exploited most fully.

The SIMKIT environment provides numerous facilities for project development such as a clock/calendar, verification procedures, data collectors, and event generators (Intellicorp, 1986). The clock, which is used to represent current simulation cycle may be associated with the calendar to schedule future events and maintain priorities (e.g., Franta, 1977). This combination will support run time checkpoints for the model that allow step-through simulation, and predetermined stop points. A hierarchical design for model development is used to promote inheritance and provide general structure verification procedures for constraint violations which include basic checks for cardinality faults, imcompatible value classes, and undefined methods. Mechanisms for dynamic, static, or stochastic data collection may be used to review, compare, and analyze the activities of the sample simulations from the data produced by event generators. These generators introduce variability into simulation through generation of pseudo-random sequences of test data.

In addition to the features provided in SIMKIT, specialized graphics will have been created earlier in the Prediction Mode for displaying sensor readouts during Diagnosis, (see Figure 5). In addition, the expert will be able to observe directly the consequences of any process rules defined during the Prediction Mode. If at any time the expert perceives the process to be going awry, s/he can stop it immediately and redesign as necessary. Such redesign could include any number of measures such as adding, deleting, or rearranging sensors, rules, or objects.

Note that the form of query in the Diagnosis Mode is more implicit than in the previous modes. There is no program of queries to which particular types of responses are expected. That is, the expert is not being given direct questions about the structure of the process. Rather, s/he is allowed to observe a step-through "simulation" as defined by information provided thus far, and then make determinations about how this design could be improved.

It should be reiterated that the Diagnosis Mode does not employ simulation in the traditional sense of the word. Rather, it is a facility in which a progression of previously defined states is displayed through, among other things, changes in sensor icons. In order to have a true simulation of the process, one would first need a structural model on which to build. Such a model may indeed be the final output of KAT, but would not be specified by the knowledge structure at this point.

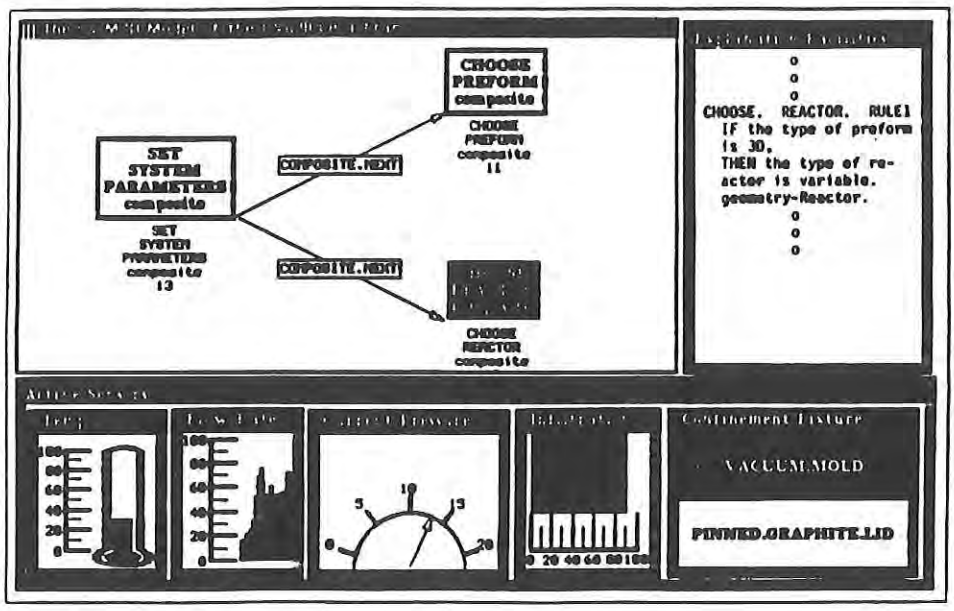


Figure 5 KAT Interface for Diagnosis Mode

SUMMARY AND CONCLUSIONS

KAT has been designed to address many types of problems typically encountered during knowledge acquisition. The fundamental feature of this tool is that it provides a variety of ways in which a knowledge engineer can query a domain expert about a materials process. Some of these queries are explicit, such as having the expert create an AND/OR graph of the model. Others are more implicit, such as letting the expert view a dynamic step-through of the process and make suggestions for redesign. It is anticipated that this combination of retrieval cues in a structured environment will facilitate the construction of knowledge bases to be used for expert system development.

This research was partially supported by the US Defense Advanced Research Projects Agency and the Army Research Office under contract number DAAL03-87-C0019.

References

Baskaran, V., & Reddy, Y.V. (1984) An introspective environment for knowledge based simulation. In Proceedings of the 1984 Winter Simulation Conference, 645-651.

Bobrow, D. (1985) Qualitative Reasoning About Physical Systems, MIT Press, Cambridge, Mass.

De Kleer, J. (1984) How circuits work. Artificial Intelligence, 24. Reprinted in Bobrow, 1984.

Franta, W.R. (1977) The Process View of Simulation, North Holland, New York.

Graf, P., & Schacter, D. (1985) Implicit and explicit memory for new associations in normal and amnesic subjects. Journal of Experimental Psychology: Learning, Memory, & Cognition, 11, 501-518.

Intellicorp (1986) The SIMKIT System: Knowledge-based Simulation Tools in KEE, Document NO. 1-1-USK-1, Intellicorp, Mountain View, CA.

Kolers, P.A., & Roediger, H.L. (1984) Procedures of mind. Journal of Verbal Learning and Verbal Behavior, 23, 425-449.

Kuipers, B. (1984) Commonsense reasoning about causality: Deriving behavior from structure. Artificial Intelligence, 24. Reprinted in Bobrow, 1985, 169-204.

Mihram, G.A. (1973) Some practical aspects of the verification and validation of simulation models. Operations Research Quarterly, 23, 1, 17-29.

Miller, D.R. (1974) Sensitivity analysis and validation of simulation models. Journal of Theoretical Biology, 48, 345-360.

Payne, J.A. (1982) Introduction to Simulation: Programming Techniques and Methods for Analysis, McGraw-Hill, New York.

Reddy, Y.V., Fox, M.S., & Husain, N. (1986) The knowledge-based simulation system. IEEE Software, 3, 2, 26-37.

Roediger, H.L., & Blaxton, T.A. (1987) Retrieval modes produce dissociations in memory for surface information. In D.S. Gorfein and R.R. Hoffmann (Eds.) Memory and Learning: The Ebbinghaus Centennial Conference, Erlbaum, Hillsdale, N.J.

A KNOWLEDGE ACQUISITION TOOL FOR THE INTELLIGENT PROCESSING OF MATERIALS

Dr. Brian G. Kushner and Dr. Roger A. Geesey
The BDM Corporation
7915 Jones Branch Dr.
McLean, VA 22102

and

Dr. Phillip A. Parrish and Maj. Steven G. Wax
Defense Advanced Reseach Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

Abstract

Today, knowledge acquisition remains the critical bottleneck in the development of expert systems. This paper summarizes research in progress to develop a tool for prototyping knowledge bases for the specialized domain of materials science. The targeted application is the integration of expert systems, in-situ sensors and process models in materials manufacturing environments.

A KNOWLEDGE ACQUISITION TOOL

FOR THE

INTELLIGENT PROCESSING OF MATERIALS

The field of expert systems is a discipline within applied AI in which human expertise in specialized domains is implemented in computer systems. These systems achieve high levels of performance in task areas that, for human beings, require years of special education and training. Obviously, the generation of a computer program with those capabilities involves an extensive development process. Although this process used to take on the order of 10 - 20 man years of effort, it has been greatly reduced by the advent of second and third generation expert system development tools. Aside from offering multiple forms of knowledge representation and control structures, these tools provide advanced development environments with such facilities as graphics, windowing, and interactive debugging systems.

Despite these recent advances in expert system technology, there is still a severe problem in building expert systems at the present time, namely the knowledge acquisition bottleneck. By knowledge acquisition, we refer to the process whereby critical information is extracted from an expert, or group of experts, and explicitly represented in the knowledge base of the expert system. There are many problems associated with this enterprise, some of which are under the control of the knowledge engineer and others which are not. First and foremost among these difficulties is the availability of and quality of time of interaction with the expert(s). Other types of challenges in knowledge acquisition include: resolving conflicts between experts, organizing and coding the knowledge obtained, and compiling the relevant expertise into problem-solving capabilities. In the following discussion we will give an overview of our recent progress to develop a Knowledge Acquisition Tool (KAT) for the Intelligent Processing of Materials (IPM) Program.

Interactive knowledge acquisition tools, such as TEIRESIAS [1], have proven their usefulness in helping the domain expert express knowledge. However, the quantity of time required by the expert still represents a significant barrier to expert systems implementation. More sophisticated systems have been built which can infer rules from presented data (e.g., MetaDendral [2]) or which can learn to guide their own search strategies (e.g., ACT* [3] and EURISKO [4]). But the principals of learning and adaptation are

still poorly understood, [5] making these systems higher risk approaches to development of functional knowledge bases. What is required for a manufacturing-based program such as IPM is an approach to knowledge acquisition which can take advantage of commercially available technologies and software, and quickly integrate them to express knowledge in an easily modified format. KAT, targeted for the IPM program, is such a tool for rapidly prototyping knowledge bases for the specialized domain of materials science.

The IPM Program seeks to combine the use of expert systems, advanced in-situ sensors, and process models to control the quality of materials in a production environment. As part of this endeavor, expertise will be acquired from multiple experts: a laboratory materials scientist, on-line technicians who possess expertise in manufacturing activities and materials applications engineers. The coupling of inputs from these very different domains is a formidable knowledge acquisition task. One can easily imagine that the knowledge engineer will have great difficulty in resolving conflicts or differences between these types of experts. The bottom line is that knowledge acquisition has a significant impact on the IPM Program, and this program cannot be successful without it.

The premise of our activity in building KAT for the IPM Program is that the knowledge and representations required for materials production are common across multiple materials processing domains. What this means is that the same type of representations, whether they be frames or rules or semantic nets, can be utilized to develop a "mediating" knowledge representation during knowledge acquisition. This "mediating" representation can form the basis for an interactive environment used to ease the socialization process associated with knowledge acquisition from experts.

As stated previously, knowledge acquisition is a critical activity in the IPM Program due to the different types of knowledge that need "to be represented in one expert system. Of particular importance is the laboratory knowledge, where an expert has multiple sensors which can be used to determine the characteristics of the materials during processing in very small quantities. There is also the production knowledge, where either of the following scenarios occur: measurements are taken from small samples and the results are used to make inferences about the quality of batch processes; or there are on-line sensors which provide minimal information about the associated process, in which case the operator makes inferences about the whole activity based on the reading of those sensors. Regardless of the configuration, both of these types of knowledge will be required in order for the IPM Program to be a success. However, knowledge acquisition is

currently a manpower intensive problem, frequently requiring the knowledge engineer to become expert in a domain. As is the case in a variety of AI technologies, the manpower intensive nature of the task is extremely costly and techniques to automate this activity are being sought.

We believe that KAT can work for the following reasons. First, previous success in AI have occurred when system designers focused in on a narrow domain, whether it be interpretation of spectral data (as in the Dendral project),6 the configuration of minicomputers from components (the RI project),7 or the diagnosis of diseases in internal medicine (the Mycin project).8 Materials production is similar to each of these in that the domain is suitably limited for expert system application. Second, the KAT may be developed using existing technology, being implemented on a Symbolics 3670 with the KEE 3.0 system.4 As such, the tool could aid in the long-term transition of materials technology from research into production, and allow us to predict integration problems in advance. Third, our experience with the development of other AI systems cause us to have a high degree of confidence in our ability to reduce the KAT design to practice. The knowledge structures within KEE and SIMKIT, a simulation building facility which is resident on top of IntelliCorp's KEE kernal, allow a library of objects to be readily created and modified. This provides the flexibility in the user interface necessary for rapid modifications and updates to the knowledge base, a critical element in the early development and prototyping of knowledge based systems.

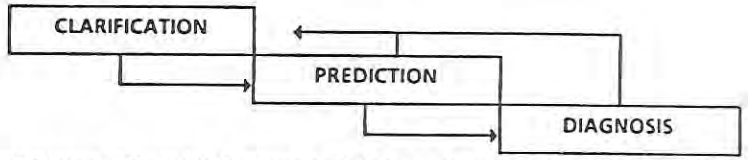
The remainder of this discussion will describe the design characteristics of KAT. Our approach to KAT includes the desire to prompt the expert's consideration of the elements that characterize his/her knowledge. That is, we seek ways to prompt the verbalization and expression of knowledge from the expert and therefore optimize the interaction time with the expert. This is done by initially structuring the knowledge engineering process through the use of a materials-referenced development environment that provides a low level of materials-related knowledge to make the expert comfortable. The expert then iterates through various stages of knowledge storage and modification which are designed to elicit extraction of different types of knowledge that she/he might have about the domain. For example, information about logical, procedural, functional and physical relations will be queried. To aid the expert, the tool serves as a workplace containing materials-specific references in easy-to-use, high graphic forms, with a facility that encourages direct query, examination through explanation, browsing, and efficient consideration of process knowledge.

To elicit knowledge, KAT is designed to present multiple, interactive displays to the expert(s) in several functional modes. These include: the clarification mode, the prediction mode, and the diagnosis mode (See figure 1). It is the iterative processing through these modes that KAT allows the expert, working in conjunction with the knowledge engineer, to initially instantiate the knowledge base and incrementally improve and modify it. The set of objects which are central to the domain under consideration are developed within SIMKIT, creating the necessary elements to explicitly represent relationships among process. For purposes of development, chemical vapor infiltration (CVI) of composite materials is the domain being addressed.

CLARIFICATION: The expert creates a graphical data-flow representation of the process.

PREDICTION:
a) The expert defines inputs to and outputs from each process.
b) Queries about process steps are answered.
c) Priority ratings for individual processes are provided.

DIAGNOSIS: The expert corrects inconsistencies in the process knowledge base through iterative exercising of system with varying inputs.



THROUGH ITERATION OF THIS PROCESS, A TOP-DOWN CONSTRUCTION OF THE KNOWLEDGE BASE IS INVOKED, SIMPLIFYING THE ORGANIZATION PROCESS

Figure 1: KAT's Functional Modes

The clarification mode allows a description of the process to be created in the form of data-flow graphs. In this mode, the expert will create a

graphical representation of the process by sketching the general process on-screen to obtain a shell to work from. This is essentially a fill-in-the-box operation (See figure 2), drawing from the set of objects previously defined within the library. The logical relations between process components are also represented, using the relations AND, OR NEXT... to explicitly define process flow characteristics.

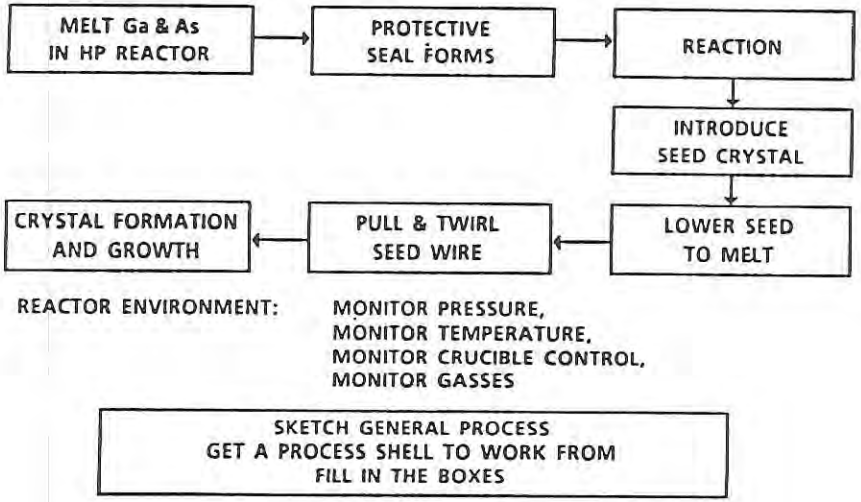


Figure 2: Clarification Mode of KAT Using Example of GaAs Bulk Crystal Growth

In the prediction mode, the expert will view the information in flowchart form, which exhibits temporal relations between process elements. The expert can elaborate on inputs to and outputs from each process (See figure 3) that was defined in the clarification mode. The expert also answers queries about active sensors, deviations from ideality, and provides priority ratings for the individual procedures. Example questions from the prediction mode are:

(1) What sensors are active at this subprocess step?
(2) What are the results if Furnace Temperature $F_T > 1000$ C?
(3) What other sensors readings would verify $F_T > 1000$ C?
(4) What other process variables elements can control F_T?

This activity might produce the following changes to the knowledge base: rearrangement of some procedures, deletion or addition of some procedures, or creation of a network effect associated with the individual processes that are represented internally.

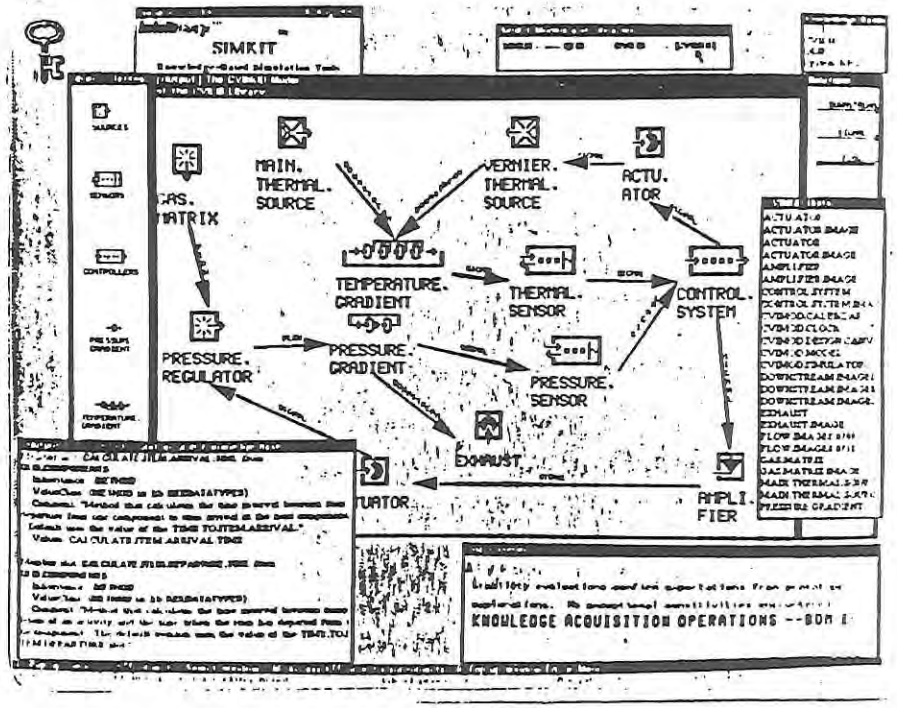


Figure 3: Prediction Mode of KAT Using Example of CVI Process

After cycling through the clarification and prediction modes several times, the expert will then move to the diagnosis mode. Here the expert provides guesses as to the causes of problems indicated in a simple simulation of the process and proposes solutions. For example, the expert might view a simulation proceeding through the flowchart previously defined

in the clarification mode. Each procedure is highlighted as it becomes active, and gauges for the relevant sensors come into view. Explanation information may be provided in a separate window and when the expert is given an undesirable result, she/he can make recommendations for adding, deleting, or changing the procedures within the knowledge base. As indicated earlier, the overall design approach for this tool involves an iteration of the process, using top down construction of the knowledge base from the clarification mode through the diagnosis mode and having several iterations in order to provide an interactive debugging facility.

Obviously, there are some trade-offs in building a system such as this. By design, KAT will have only a sophomore's level of understanding, whereas a fully functional expert system should by definition have an expert level of understanding about the process. But, there lies the difference between the mission of KAT as compared to the expert system. KAT is intended to be a supplement to, and not a substitute for, the eventual expert system. That is, KAT is a general tool seeking application in multiple domains, whereas any expert system is specialized in depth with regard to a particular set of knowledge structures. In addition to its generality, the cost of implementation for KAT will be low since only a minimum level of knowledge need be initially built into the system. Furthermore, it is designed for portability and to leverage the expert's time and availability.

In summary, the Knowledge Acquisition Tool attacks a major problem in both expert systems and the Intelligent Processing of Materials program -- the knowledge acquisition bottleneck. We believe that this tool can aid in the identification of common elements across multiple materials processing domains, facilitation the rapid implementation of materials manufacturing knowledge bases. Finally, KAT can assist in demonstrating rapid transition from materials research into production and in streamlining the development of expert systems.

Acknowledgements

The authors wish to thank Drs. T. A. Blaxton and C. B. Friedlander for their guidance and assistance in the development of KAT. The authors also wish to express their gratitude to Dr. D. Ulrich of the Air Force Office of Scientific Research (AFOSR) for enabling critical components of this research to be performed. Two of the authors (BGK and RAG) would like to acknowledge the support for a portion of this work by DARPA and AFOSR under contract number F49620-86-C-0036.

References

1. R. Davis and D. Lenat, **Knowledge-Based Systems in Aritificial Intelligence**, McGraw-Hill, New York, 1982.

2. B. G. Buchanan, D. H. Smith, et al, "Applications of Artificial Intelligence for Chemical Inference XXII: Automatic Rule Formation in Mass Spectrometry by Means of the Meta-DENDRAL Program," **Journal of the American Chemical Society**, *Vol 98* (1976) 6168.

3. J. R. Anderson, **The Architecture of Cognition**, Harvard University Press, Cambridge, MA, 1983.

4. D. Leant, "EURISKO: A Program that Learns New Heuristics and Domain Concepts," **Artificial Intelligence**, *Vol. 21, No. 1*, (January 1983) 31

5. T. A. Blaxton and B. G. Kushner, "An Organizational Framework for Comparing Adaptive Artificial Intelligence Systems," **Proceedings of the 1986 Fall Joint Computer Conference**, IEEE Press, Washington, DC, 1986.

6. R. Lindsay, B. Buchanan, E. Feigenbaum, J. Lederberg, **Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project**, McGraw-Hill, New York, 1980

7. J. McDermott, "R1: The Formative Years," **AI Magazine**, *Vol. 2, No. 1* (Spring 1981) 21

8. E. H. Shortliffe, **Computer-Based Medical Consultations: MYCIN**, American Elsevier, New York, 1976

Cognitively-Based Knowledge Acquisition: An Integrated Toolkit*

Juliana S. Lancaster

BDM International, Inc.
1300 N. 17th Street, Suite 950
Arlington, VA 22209

INTRODUCTION

As increasing numbers of knowledge-based systems are developed for operational use, the importance of knowledge acquisition becomes even more apparent. Knowledge acquisition, the process of extracting and transferring expertise from domain experts into the system, consumes the single largest block of time in the development of a knowledge based system (Feigenbaum, 1977; Hayes-Roth, et al., 1983; McGraw & Harbison-Briggs, 1989). To reduce the time required for this process, numerous tools have been developed to automate or systematize the elicitation and transfer of knowledge from expert to knowledge base. Among these advances are developments in machine learning that enable an expert system to acquire its knowledge from "experience" (Michalski, 1980; Michie, 1982), expert system shells that allow domain experts to write their own rules (e.g., MacSMARTS), and knowledge engineering tools that assist a knowledge engineer in his/her work. The material reported here describes research and development efforts on a computer-based set of knowledge acquisition tools known collectively as the BDM Knowledge Acquisition Toolkit (BDM-KAT).

Work described in this paper was supported by the US Defense Advanced Research Projects Agency, through the Air Force Office of Scientific Research under contract number F49620-86-0036 and the Army Research Office under contract number DAAL03-87-C-0019. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing the official policies, either express or implied of the Defense Advanced Research Projects Agency or the US Government.
Special thanks to R. Geesey, B. Kushner, C. Friedlander, L. Reeker, and C. Westphal for their comments on earlier drafts of this paper.

BDM-KAT differs from other tools in several critical ways. It has been designed and implemented to aid a trained knowledge engineer in eliciting expert knowledge about well-structured domains. It is based on principles of human cognition relating to the representation and structure of knowledge in memory, yet these representations are implemented in a computationally-efficient way. The toolkit provides an environment in which the knowledge engineer and domain expert discuss, manipulate, and record the objects and relationships that provide the domain with its structure and predictability. As the elicitation progresses, a database of basic domain components is developed. Using this database, a knowledge base and inferencing structure can be derived to operate in the selected domain.

The development of BDM-KAT can be discussed from several perspectives, including its conceptual basis, implementation, integration, and intended uses. The purpose of this paper is to address briefly each area with particular emphasis on the roles of a cognitive theory in BDM-KAT. First, we present the need for and define the framework upon which BDM-KAT has been designed. Second, we present the specific organizational structures reflected in the initial components of the toolkit. Finally, we describe, from a cognitive perspective, the implementation, integration and use of BDM-KAT.

KNOWLEDGE ENGINEERING METHODOLOGIES

Knowledge engineering as a field has developed from the demand created by the development of expert or knowledge-based systems. In such applications, it is necessary that the knowledge base used by the system be well-organized, consistent, and reasonably complete. This, in turn, requires that the process by which that knowledge is entered into the knowledge base be equally well-organized and controlled. However, early efforts in knowledge acquisition were modeled on small scale projects in which a single domain expert and a single developer could interact to build the final project, allowing a certain

informality and ad hoc set of procedures to exist. In larger scale projects, the ad hoc nature of early knowledge engineering efforts was soon discovered to be limiting and unworkable.

In response to this situation, Hayes-Roth, et.al. (1983) presented a methodology for knowledge engineering. In this approach, as Figure 1 shows, a five-step cycle of identification, conceptualization, formalization, implementation, and testing is used iteratively until a satisfactory product results. While this provides a structure within which a system architecture and basic domain structure can be determined and allows for iteration through an identifiable sequence, it does not impose any control or guidance on the techniques used or support verification of the elicited knowledge. More current approaches tend to depict the knowledge engineering process as a more controlled, cyclic sequence, drawing on models and standards from the software engineering field. Some methodologies, like the adapted waterfall model of knowledge acquisition presented by McGraw and Harbison-Briggs (1989), suggest not only steps within the cycle, but also techniques for each phase and some guidance at decision and iteration points. However, even this structure does little to address how the knowledge acquisition task might be tied directly to the perceived structures and representations within the domain, as viewed by an expert.

Insert Figure 1 about here

In any manual knowledge engineering effort, the time and labor demands are high even when structures and practices are in place to direct and manage the timing and foci of knowledge engineering sessions. In addition, there are several pervasive and troublesome characteristics of the overall process. Specifically, true experts, because of the demand for their expertise are frequently unavailable for a series of several

knowledge acquisition sessions, thus generating a demand on the knowlege engineer to maximize the value of every session held. When sessions are held, experts frequently have great difficulty accessing and then verbalizing significant components of their knoweldge because it has been "chunked" or "proceduralized" during the development of expertise. Such knowledge is easily acted upon, but not easily discussed, making it difficult or impossible for the knowledge engineer to tap and depict the structure and relationships the expert holds of the domain. In addition, because such knowledge is, in many ways, implicitly assumed by the expert, it allows him/her to assume a level of understanding on the part of the knowledge engineer that is frequently inappropriate. Furthermore, many knowledge engineers cling to the interview as a primary thechniques. In the hands of a skilled communicator with adequate analytical skills and a good foundation in the domain, the interview can be highly useful. However, few knowledge enigineers are able to use it efficiently, leading to large expenditures of time that result in little critical knowledge being elicited. This can be compounded by the practice of establishing the design and architecture of the target system in advance of knowledge acquisition--a practice that forces a way of thinking about the domain and subdomains that may be incompatible with the model held by the domain expert. Finally, regardless of the techniques used or time expended, many manual-based knowledge acquisition efforts fail to document the knowledge elicited sufficiently for later verification and validation of the system.

A number of approaches to automating the process so as to alleviate some of these have been tried, ranging from simple problem simulations for discussion to fully automated knowledge acquisition environments. Most of these knowledge engineering tools have been developed to serve the needs of a specific system or a particular class of applications. In other words, they can be considered to be *implementation driven*. Because of their orientation toward the final representation and use of the knowledge by the

system to be developed, the tools constrain domain experts by requiring that they present knowledge in the format on which the tool is based (e.g., repertory grids, fuzzy sets) or in the representation of the knowledge base (e.g., rules). This requirement may force the experts to transform the expression of their knowledge from a familiar, useful form to a novel, untested one. In contrast, many manual approaches do allow the expert to express knowledge in a more comfortable manner but place the burden for the transformation on the knowledge engineer. In designing BDM-KAT, we first considered what would be desired characteristics of an improved knowledge engineering environment and sought to develop an approach in line with them. Briefly, we determined that a knowledge engineering system should:

- reduce the requirements for translations of knowledge, particularly by the domain expert;
- reduce the burdens in time and effort on the domain expert;
- reduce the dependence on the skill of the individual knowledge engineer;
- provide a mechanism for selecting appropriate techniques; and
- provide an overall control structure for a knowledge elicitation effort.

In BDM-KAT we seek to meet these goals by using a theoretical foundation from cognitive psychology (Vekker, 1974; 1976; 1981) and by controlling knowledge elicitation through the use of recognized conceptual mechanisms such as categorical structures (Rosche, Mervis, Gray, Johnson, & Boyes-Braem, 1975), causal models, and timelines of events (Barsalou, 1986). In the next section, we present some details of the cognitive framework that has driven the design of BDM-KAT; after that we discuss the characteristics of human knowledge organization and reasoning that have been central to BDM-KAT thus far.

A MULTI-DIMENSIONAL MODEL OF HUMAN KNOWLEDGE

A first premise of BDM-KAT is that all aspects of an expert's knowledge must be captured in a knowledge base for an intelligent system. Equally important is the premise that the knowledge structures as elicited and represented should be compatible with the structures used by the expert. There is substantial agreement with both of these premises within the AI and cognitive science communities (Barfield, 1986; Gammack, 1987; Hamill, 1984; Kidd, 1983; McGraw & Harbison-Briggs, 1989). However, as discussed above, the problem of eliciting and representing knowledge is highly complex. When compatibility with human knowleldge structures is made a priority, it becomes necessary to address the even more perplexing issue of non-verbal knowledge. This presents an extremely difficult situation, particularly because the nature of non-verbal knowledge is not well understood by cognitive psychologists. Generally speaking, the cognitive aspects of non-verbal knowledge have been treated in more detail by Eastern psychologists than those of North America, where there is still controversy about the reality of direct visual representation (represented by the advocacy of dual coding by Paivio (1986) and its rejection by Pylyshyn (1984)). Within the work developed overseas, the work of Provessor L. M. Vekker (1974, 1976, 1981) is outstanding. Professor Vekker has developed a theory of human knowledge processing that integrates all levels of mental activity from perception to abstract thought. In the following paragraphs, we provide a very brief summary of those aspects of Vekker's theories that have played roles in the design of BDM-KAT.

Vekker has developed, over a research career of 40+ years, a comprehensive theory of the human cognitive system. He drew strongly on the general structure of information theory (see Billingsley, 1963 for definition and discussion of the principle concepts in information theory) in building his multi-dimensional model of the human system. In particular, he defines knowledge as a special type of information within that framework and uses the principles of isomorphism and homeomorphism, or the

one-to-one mapping of different functions or knowledge representation to one another to develop his model of the levels and types of knowledge within the human system and their relationships to one another. Vekker describes the internal store of information as being of two fundamentally different types (imagistic and sentential) that are reflected in multiple levels of the cognitive system. Figure 2 summarizes the types of knowledge and the increasing levels of abstraction discussed in Vekker's model. Perceptual and memory "images" reflect some specific object or experience or a hierarchically abstracted, idealized image of a class of objects or experiences such as a prototypical object. Images include not only *visual* images but also such sensation-oriented representations as the *kinetic* sense of balance and muscle motion required to ride a bicycle and the *haptic and kinetic* awarenesses of when a football is held and thrown properly. In contrast, a sentential representation is not tied to any specific experience; it is an abstract and verbally-oriented representation. Since the imagistic components of awareness of specific and generic objects and the sentential concepts of names of such objects are represented and stored separately, thinking or reasoning processes require the manipulation of connections that exist between sentential represenations and their related imagistic representations in a process of *continuous reversible translation*, as indicated in Figure 3.

Insert Figure 2 about here

Insert Figure 3 about here

Dr. Vekker also discusses learning and representation in individual and multiple modes. While the knowledge or information necessary for reasoning in a domain is being

learned, the learner will have many inconsistencies in her knowledge, particularly between imagistic and sentential representations. In contrast, experts tend to have developed stable knowledge representations that allow them to manipulate the knowledge to make inferences, to relate similar knowledge, and to be consistent in their statements, assumptions, and conclusions. The degree of instability in a student's problem solving or question answering, where manipulation of sentential and imagistic representations simultaneously is required, is a measure of the degree to which she has not yet mastered the material to be learned. Professor Vekker emphasizes in his discussions of knowledge elicitation that questions and problems must be posed to the expert that will drive the interaction of sentential and imagistic knowledge so that stability in the translations between the two "languages" can be demonstrated. Only then can the elicited knowledge be considered valid and acceptable across modes. In BDM–KAT, the methods and procedures for using the tools have been developed to encourage the expert to draw upon his/her imagistic and sentential representations to enable identification of the critical invariant relationships in the expert's knowledge. Critical invariants are those aspects of a concept or causal sequence that cannot be altered without substantively changing the nature of the concept. For example, a (American) football game has numerous characteristics, including the size of the field, shape of the goal posts, nature of the uniforms worn, and time limits on play that, if violated, do not prevent the ongoing activity from being considered a football game, albeit a "back yard game" or "pick up game" or some other informal variant. Nevertheless, if other characteristics of the game such as the shape of the ball or the means by which the ball is placed into play or transferred from one player to another, then what is played is *not* a football game at all, but another game altogether, perhaps soccer. This exercise indicates that while those characteristics in the first list are *important* features of a game, they are not *critical invariants* of the general concept. However, if the concept is limited to an officially

sanctioned inter-mural game, then those characteristics may well turn out to be critical invariants of the more limited concept.

Several characteristics of BDM-KAT are the direct result of this cognitive framework. In the current version of BDM-KAT, the interface representations have been designed to take advantage of the verbal and nonverbal aspects of the knowledge structures being elicited and the initial set of tools implemented has been selected to draw upon knowledge structures common to all human information processing. In addition, an overall integration schema for the toolkit has been partially developed that includes additional tools and a "manager" component to assist the knowledge engineer in coordinating sessions and integrating the knowledge base. The set of tools that we expect to eventually comprise the complete toolkit will reflect the cognitive system as shown in Figure 4. Briefly, the functions of the tools are (or will be) as follows:

Categorizer: elicits basic domain concept structure in rigid class/subclass hierarchies and tracks concept properties.

Classifier: acts as Categorizer, but also supports creation of tangled concept hierarchies.

Time Mapper: elicits temporal sequences of events, supports decomposition of processes into event/subevent hierarchies, tracks temporal characteristics of events.

Causal Charter: elicits causally connected sequences of events, states, and concepts.

Imager: supports input of graphic images and elicitation of information implicit in the images.

In a later section of this chapter, the implemented tools are discussed in some detail.

Insert Figure 4 about here

COMMON KNOWLEDGE STRUCTURES

This framework of the ways in which human knowledge is structured along with our knowledge of how such structures are organized has driven the selection and design of the initial set of tools for BDM-KAT. Expertise can rarely be elicited and represented in a single structure as humans tend to store knowledge about a domain in multiple structures. Knowledge engineers, and tools developed to assist in the process of knowledge acquisition, must be capable of eliciting, translating, and representing multiple types of knowledge structures for any given domain. Complex, knowledge-intensive domains reflect diverse, yet interrelated types of knowledge. For the initial set of tools in the BDM-KAT prototype, we have focused on those recognized to be universally applied to verbal knowledge, specifically categorical, temporal, and causal structures. The following paragraphs present basic information about the nature and support for the generality of these structures.

Categorical Structure of Knowledge

People tend to divide the world into relatively neat categorical structures. While for laymen the categorical divisions are frequently unstable (e.g., individual items may change their positions in terms of both category membership and typicality within a category (Barsalou, 1986)), experts in a specific field tend to hold stable and shared categorizations of their domain (Rosche, et. al. 1975). A substantial portion of one's understanding of a domain rests in knowledge about the items that appear in the domain, their individual and collective characteristics, and the relationships between them. Consider again, for example, our football game and its position in a general categorical

structure of games. Few (even non-experts) would have trouble producing a taxonomy such as that shown in Figure 5 and naming the characteristic properties of each node. Undoubtedly, a sports professional would quickly generate a more detailed, and possibly differently structured taxonomy. Because this organization is natural to the domain expert, knowledge elicited within this structure is likely to be more easily obtained, more easily modified, and, once accepted by the expert, more accurate.

Insert Figure 5 about here

Temporal Organization of Events

In addition to dividing objects in the world into categorical sets, people structure their knowledge and memories of events into orderly sequences. Autobiographical knowledge appears to be organized internally into "mental time line models" in which the temporal relationships between episodes are specified (Barsalou, 1986). Similarly, individual events and generalizations of events appear to be stored menally as ordered sequences of actions and/or decisions, such as those in the familiar restaurant script, doctor script, and so forth (Mandler, 1979; Nelson, 1978; Schank & Abelson, 1977). Where the episodes or process events can be decomposed to enabling or sub-events, hierarchical relationships exist between them similar to those between levels in a categorical taxonomy. This structural nature of event memory can be exploited in eliciting knowledge about processes that occur over time. Drawing from this model the domain expert can tap and share the order, temporal characteristics and relationships, and hierarchical associations between steps in the process.

Causal Structures in Knowledge

Finally, experts can be partially defined as those individuals who understand the *why* and *how* of a particular domain. All people, even novices, construct causal mental models of the world around them (Kelly, 1955). However, causal models held by a novice may reflect incorrect assumptions or focus (e.g., correlations), while those held by an expert tend to be more accurate. Consider, for example, the knowledge used by auto mechanics. While they do develop, with expertise, extensive classifications of diagnoses by symptom sets, they continue to solve most diagnostic problems by tracing the presented malfunction through its causal antecedents to (hopefully) a single cause. The networks of causal connections supporting this reasoning have been shown to improve in specifiable ways with increasing experience and training (Lancaster & Kolodner, 1987; 1988). As with categorical and temporal organizations of knowledge, causal models can be elicited directly from an expert--here by focusing interactions on the expert's explanations for states and conditions in the domain of interest. Of these three types of knowledge, causal models are the most similar to the rules usually sought during the knowledge acquisition phase of knowledge-based system development.

BDM-KAT DESIGN AND IMPLEMENTATION

BDM-KAT is a toolkit that has been designed to enable a knowledge engineer to elicit knowledge based on both the structure and modality in which it is most likely to be represented in the domain expert's internal cognitive model. To accomplish this, BDM-KAT consists of three cooperating knowledge engineering tools: Categorizer, Time Mapper, and Causal Charter. Their high-level structure is shown in Figure 6. *Categorizer* elicits knowledge about the stable, underlying structure of a scientific domain via the drawing and "filling in" of a tree-like, taxonomic structure. *Time Mapper* provides a context in which the domain expert positions items on time lines to express knowledge about the temporal characteristics of, and relationships between, events and subevents within a process. The knowledge engineer uses *Causal Charter* to elicit cause-effect

knowledge for domain functions and events. Each tool provides the knowledge engineer and domain expert with a means for 1) focusing attention on the aspect of the domain being elicited, 2) recording information as it is elicited, and 3) verifying the information as it is elicited.

Insert Figure 6 about here

Each tool uses textual and graphic interface representations, and all tools transfer information to, and interact with, a single database management system (DBMS). The final database representation of the knowledge has three potential uses. First and formost, the database should be of form and content sufficient to serve as an intact knowledge base for insertion into a knowledge-based system. Secondly, the structure of the database itself should provide significant support for the generation of knowledge documents, which are on-line or hard-copy structured summaries intended for review, verification, and validation activities by the domain expert, knowledge engineer, and other system development team members. Finally, because the knowledge will be stored in a supported DBMS, it will be possible for human users of the knowledge to query and browse the content of the database for assistance in decision making outside of the context for which the knowledge-based system has been developed.

The following sections briefly describe the individual tools. This version of BDM-KAT has been successfully demonstrated on several occasions and has been used to elicit a significant knowlege base of HIP information.

Categorizer

Categorizer elicits the underlying structure of a domain by requiring the expert to construct a taxonomy or set of taxonomies. As discussed earlier, categorical knowledge is

commonly held for any domain or topic area. Furthermore, it is critical for constructing a domain model because the properties of specific objects and object classes will dictate much about how the knowledge can be used.

Using Categorizer the expert initially creates and names objects that exist in the taxonomy and places them at the appropriate position within the structure. The expert typically works in a top-down fashion, but the tool does not restrict the level at which objects can be named. After the taxonomy has been developed the essential, contingent, and second-order properties of each domain object are elicited from the expert. (A clear knowledge of object properties is necessary for the success of the overall knowledge engineering effort because it will be these properties that will determine how the model can be perturbed during future verification activities.) Object properties are best elicited in tandem with object taxonomies for two primary reasons. First, this sequence follows from the structure of the taxonomy itself. Second, while the expert's attention is focused on the taxonomy, s/he will be less likely to make errors in identifying properties for individual objects. Figure 7 shows a partially completed categorical elicitation.

Insert Figure 7 about here

Categorizer creates and builds within the database a series of entities and relationships between them. The domain objects and properties specified during elicitation are represented individually in the database as are their relationships.

Time Mapper

Whereas Categorizer elicits taxonomic and part-whole relationships among objects, Time Mapper elicits the structure of events or actions that occur during the execution of a process. Specifically, Time Mapper elicits the hierarchical relationships

among events and subevents (at multiple levels), the overall duration of each event, and the specific starting and ending times of each subevent within an event. This elicitation effort is critical to developing a full process model because it provides both the decomposition of the process into well specified components and the temporal relationships necessary for control. The expert is asked to name the process as a single event and then to name the subordinates of that event. This cycle iterates with increasing specificity, such that the subevents at one level are considered as events at the next level, until the events can be decomposed no further. Figure 8 shows a partially completed time map.

Insert Figure 8 about here

Causal Charter

The third tool, Causal Charter, is used to elicit high level "rules" and low level cause-effect chains. The rules are intrinsically implicit and are supplied by the expert through a description of an item to be "charted". The item under consideration may be either a function of a domain object (specified during Categorizer as a functional property of that object) or a subevent from the process decomposition state of Time Mapper. The structural elements in the item description are separated from other specified functions and are grouped together to indicate those elements that support each function in the description. Thus, the generation of clusters of structural elements (as units) are related to specific functions. The expert is asked to select those items from the description that provide support for each function given in the description. When specifying the cause-effect chains the domain expert selects the elements from the mechanism *in the order in which they are caused* or required by the mechanism. Thus a completed chain indicates the process by which the final state or element is reached. Causal Charter can

also serve as a verification mechanism on some components of the categorical and process knowledge elicited previously.

Scenario Using BDM-KAT

Given this description of the general functionality of each tool, let us now consider their coordinated use in a knowledge acquisition effort. As with any such exercise, the knowledge engineer's first task is an organizational one; in this case identifying the types of knowledge to be elicited and selecting the tool to support each session. For example, suppose a knowledge engineer were tasked to structure a knowledge base for a software engineering support tool. Analysis of the task domain would allow the knowledge engineer to determine that the tool will need to have two primary forms of knowledge in its knowledge base: that about the timing and sequence of various planning, documentation, code development and testing activities leading up to a final software product; and that about the set of documents and document components required for the developing and final software product, their characteristics, and their relationships to one another and the steps in the process. From this characterization of the knowledge base requirements, the knowledge engineer concludes that the first of these classes of knowledge can be elicited using the Time Mapper tool and the second using the Categorizer tool. Because the process of software development is only tangentially driven by direct causal factors, the knowledge engineer makes no initial plans to use the Causal Charter tool during eliciation activities.

Based on preliminary conversation and scheduling with the domain expert, the knowledge engineer decides to begin the elicitation activities with a series of sessions using the Time Mapper tool. This decision is driven by the domain expert's statement that s/he will find it easier to identify all the document requirements from their relevant points in the process and sense that the process sequence may be the more straightforward to express, at least in an idealized form. In the first session, the expert

and knowledge engineer focus on establishing the overall time frame (months or years) for developing a large-scale software product and on establishing the general stages through which such a project should move. During discussions of timing factors, the knowledge engineer focuses the domain expert on how much time is allocated to each phase of the process and how much they can or should be allowed to overlap one another. In subsequent sessions, each phase of the process is broken down into smaller steps and the timing and sequencing of the steps discussed until the domain expert states the the tasks are defined as finely as possible or as finely as is appropriate for consideration by the support tool. At the end of each session, the domain expert is provided with a hard copy of all the material discussed during the session, and all material discussed at previous sessions to support review and correction of already elicited material and the expert's own preparation for the next session. In particular, the expert is asked to prepare for subsequent sessions by considering the next level of task breakdown.

With the process sequence and timing structured to the expert's satisfaction, the knowledge engineer is ready to turn to a series of sessions using the Categorizer tool. In preparation for the first session, the domain expert is asked to review the process sequence and to note down document requirements that are linked to specific points in the process. In addition, the domain expert is asked to reveiw or to bring any support materials that define the structures of the documents insofar as they can be separated into independently prepared components. Armed with this information, the domain expert and knowledge engineer begin their first session with the Categorizer tool by enumerating the set of documents required and placing them into a hierarchical structure that, at its highest level, reflects the general phases of the development process.[1] For each document, the domain expert is asked to specify the properties of the document.

1 While, in its current implementation, BDM-KAT does not directly link the Categorizer information to the events defined in Time Mapper, having the process structure available during the Categorizer elicitation does assist the knowledge engineer and expert in defining the hierarchical relationships and, in BDM-KAT's final form, the connections would be generated and maintained in the knowledge base.

Properties, in this domain would include such things as the purpose of the document and any official numbering conventions for document identification. With the set of full documents defined and described, the expert and knowledge engineer thed proceed to address the lower level sections and subsections of each document, placing them into their appropriate relationships to the parent documents and specifying their characteristic properties until the documents have been defined to the finest level at which a section could be generated somewhat independently of other sections.

This brief review of the use of BDM-KAT has demonstrated that the tools constructed map well into the types of knowledge required by knowledge based systems of several types. In addition, the knowledge engineer and expert are aided in both preparation for and execution of elicitation sessions by their knowledge of how the tool will be used to structure and focus the discussion for each session and by the easy availabilty of hard copies of the material for review and modification purposes.

CONCLUSIONS

BDM-KAT is a cognitively-based toolkit that represents a major shift in the development of tools to support the knowledge acquisition process. While previous tools assisted the knowledge engineer in efforts related to programming or implementation, BDM-KAT has been developed based as an application of cognitive research to assist the knowledge engineer in eliciting and representing knowledge in a cognitively valid, computationally efficient manner. We believe BDM-KAT has the potential to provide significant assistance to knowledge engineers in three broad areas. First, BDM-KAT can streamline the overall process by:

- representing objects at the appropriate conceptual level;
- enabling elicitation and representation of fine-grained temporal knowledge;

- enabling direct elicitation of causal knowledge;
- capturing relevant heuristic knowledge provided during other sessions on-line; and
- using a high-quality, friendly user interface.

In addition, BDM-KAT will improve the nature of the domain expert's interactions during knowledge elicitation by:

- providing a high-quality interface;
- supporting verbalization of knowledge by the expert via cognitive compatibility;
- supporting more focused and efficient sessions; and
- supporting immediate print-out and review of elicited knowledge.

Finally, BDM-KAT enhances the verifiability of the elicited knowledge by:

- supporting immediate print-out and review of elicited knowledge;
- allowing on-line browsing of all elicited knowledge at any stage of elicitation;
- using cognitively compatible interface representations; and
- reducing the amount and number of "translations" between the expert and the knowledge base.

In addition to enhancing the quality and timeliness of the knowledge acquisition process, BDM-KAT will generate a knowledge base whose contents reflect the expert's mental organization. Such a structure is most likely to result in a knowledge-based system, intelligent tutoring system, or decision aiding system that can be used effectively in operational settings. It is our position that the cognitive foundations of BDM-KAT and its orientation toward the acquisition of knowledge in a form familiar to and comfortable for the domain expert have resulted in a system whose utility is not constrained to a specific domain or domain class.

The overall concept of BDM-KAT calls for a fully integrated set of tools and a coherent, integrated database representation of the elicited knowledge to which inferencing techniques can be applied. Research on knowledge acquisition in the context of *BDM-KAT* development is continuing with the emphasis moving toward developing the capabilities that will realize the toolkit's potential for general purpose use in three broad technical areas of research:

- visualization of knowledge for elicitation and control purposes;
- integration of the elicited knowledge and the BDM-KAT tools; and
- interfaces between the environment and its users.

These topics will be pursued and new tools and capabilities implemented as the appropriate designs are developed. In all continuing work, the predictions and parameters of the human cognitive systems, as modeled in the multi-dimensional model discussed earlier will be used to define boundaries and requirements of the system.

REFERENCES

Anderson, J.R. (1984). *Representational types: A tricode proposal.* Technical Report #82-1. Office of Naval Research, Washington, D.C.

Barfield, W. (1985). Expert-novice differences for software: Implications for problem-solving and knowledge acquisition. *Behavior and Information Technology, 5(1).*

Barsalou, L.W. (1986). The content and organization of autobiographical memories (unpublished draft).

Billingsley, P. (1963).

Feigenbaum, E. (1977). The art of AI. *Proceedings of the Fifth International Conference on Artificial Intelligence.*

Gammick, J.G. (1987). Modelling expert knowledge using cognitively compatible structures. In: *Proceedings of the Third International Expert Systems Conference*: Oxford.

Hamill, B.W. (1984). Psychological issues in the design of expert systems. In: *Proceedings of the 28th Annual Meeting of the Human Factors Society*: San Antonio, TX.

Hayes-Roth, F., D. Waterman, & D. Lenat (1983). *Building Expert Systems* Addison-Wesley.

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

Kidd, A.L. (1983). Human Factors in expert systems. In K. Coombes (ed.) *Proceedings of the Ergonomic Society Conference:* Taylor and Francis, London.

Lancaster, J.S. and J. L. Kolodner. (1988). Varieties of learning from problem solving experience. In: *Proceedings of the Tenth Annual Conference of the Cognitive Science Society.*

Lancaster, J.S. and J. L. Kolodner. (1987). Problem solving in a natural task as a function of experience. In: *Proceedings of the Ninth Annual Conference of the Cognitive Science Society.*

Mandler, J. (1979) Categorical and schematic organization in memory. In C.R. Puff (ed) *Memory Organization and Structure.* New York: Academic Press.

McGraw, K. and K. Harbison-Briggs (1989). *Knowledge Acquisition: Principles and Guidelines.* Englewood Cliffs, NJ: Prentice-Hall.

Michalski, R. (1980). Knowledge engineering through conceptual clustering: A theoretical framework and algorithm for partitioning data into conjunctive concepts. *International Journal of Policy Analysis and Information Systems, 4(3),* 219-243.

Michie, D. (1982). The state of the art in machine learning. In D. Michie (ed) *Introductory Readings in Expert Systems.*

Nelson, K. (1978). How children represent knowledge of their world in and out of language: A preliminary report. In R.S. Siegler (ed) *Children's Thinking: What Develops?*. Hillsdale, NJ: Lawrence Earlbaum, Associates.

Paivio, A. (1986). *Mental Representations: A Dual Coding Approach*. New York, Oxford University Press.

Pylyshyn, Z. W. (1984). *Computation and Cognition: Toward a Foundation for Cognitive Science*; MIT Press, Cambridge MA.

Rosche, E., C.B. Mervis, W.D. Gray, D.M. Johnson, & P. Boyes-Braem (1975). Basic objects in natural categories. *Cognitive Psychology, 8*, 382–439.

Schank, R. C. and R. Abelson (1977) *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Lawrence Earlbaum, Associates.

Vekker, L.M. (1981) *Mental Processes: The Agent, the Experience, Action and Consciousness* Vol. 3, Leningrad University Press. (In Russian).

Vekker, L.M. (1976) *Mental Processes: Thinking and the Intellect*. Vol. 2, Leningrad University Press. (In Russian).

Vekker, L.M. (1974) *Mental Processes*. Vol. 1, Leningrad University Press. (In Russian).

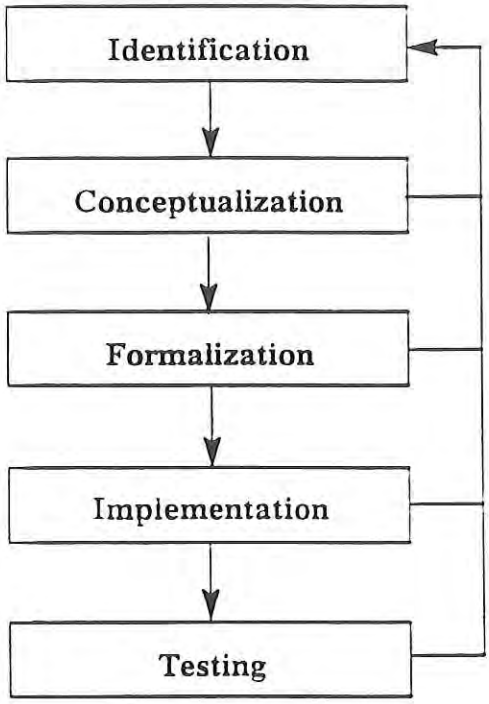


FIGURE 1 INERATIVE MODEL OF KNOWLEDGE ENGINEERING

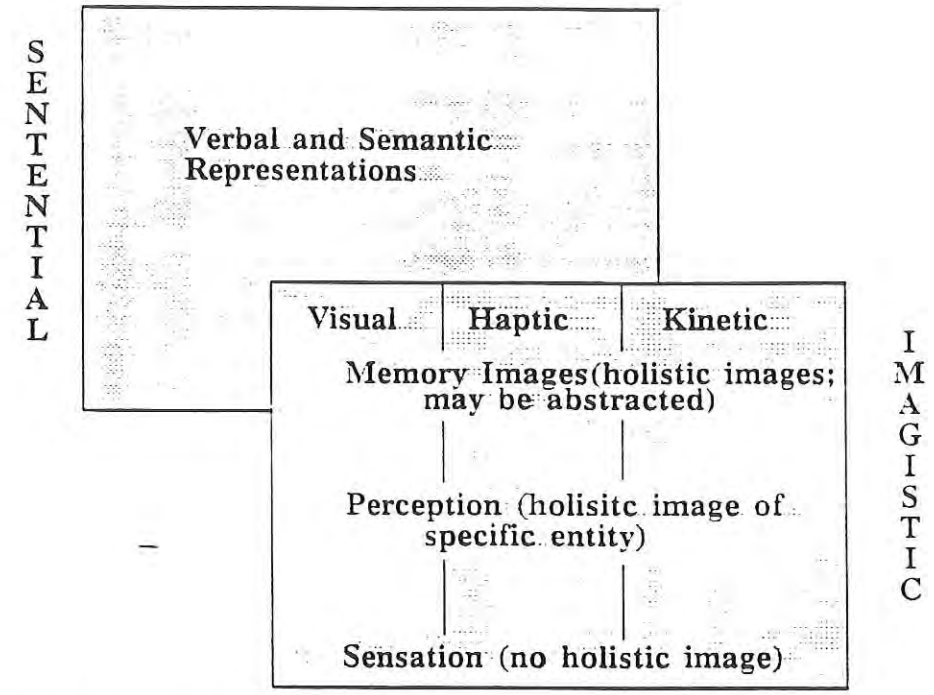


FIGURE 2 REPRESENTATIONS IN THE HUMAN COGNITIVE SYSTEM

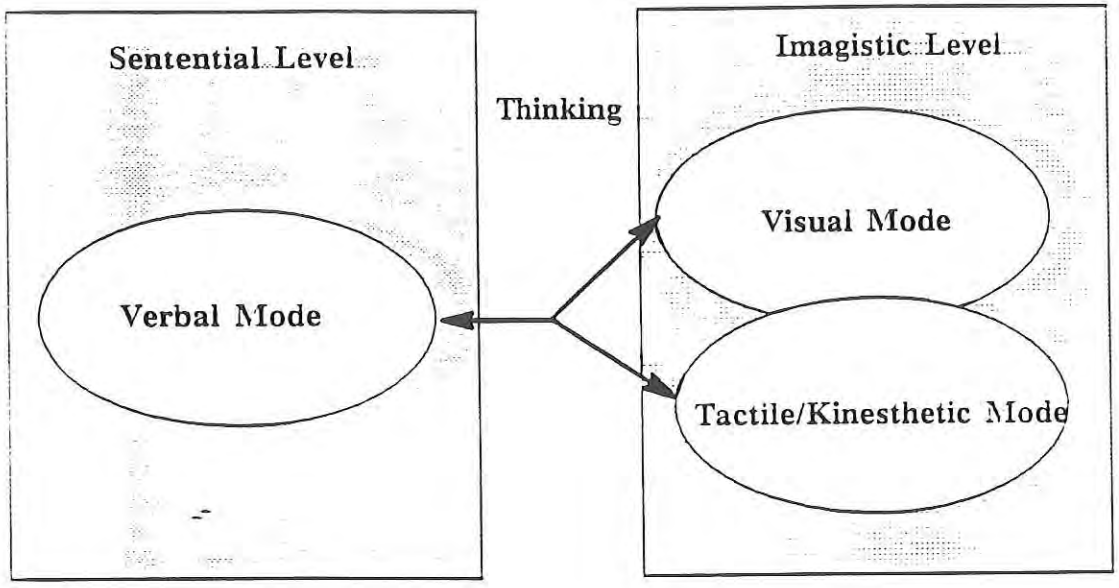


FIGURE 3 INTERACTIONS IN THE COGNITIVE SYSTEM

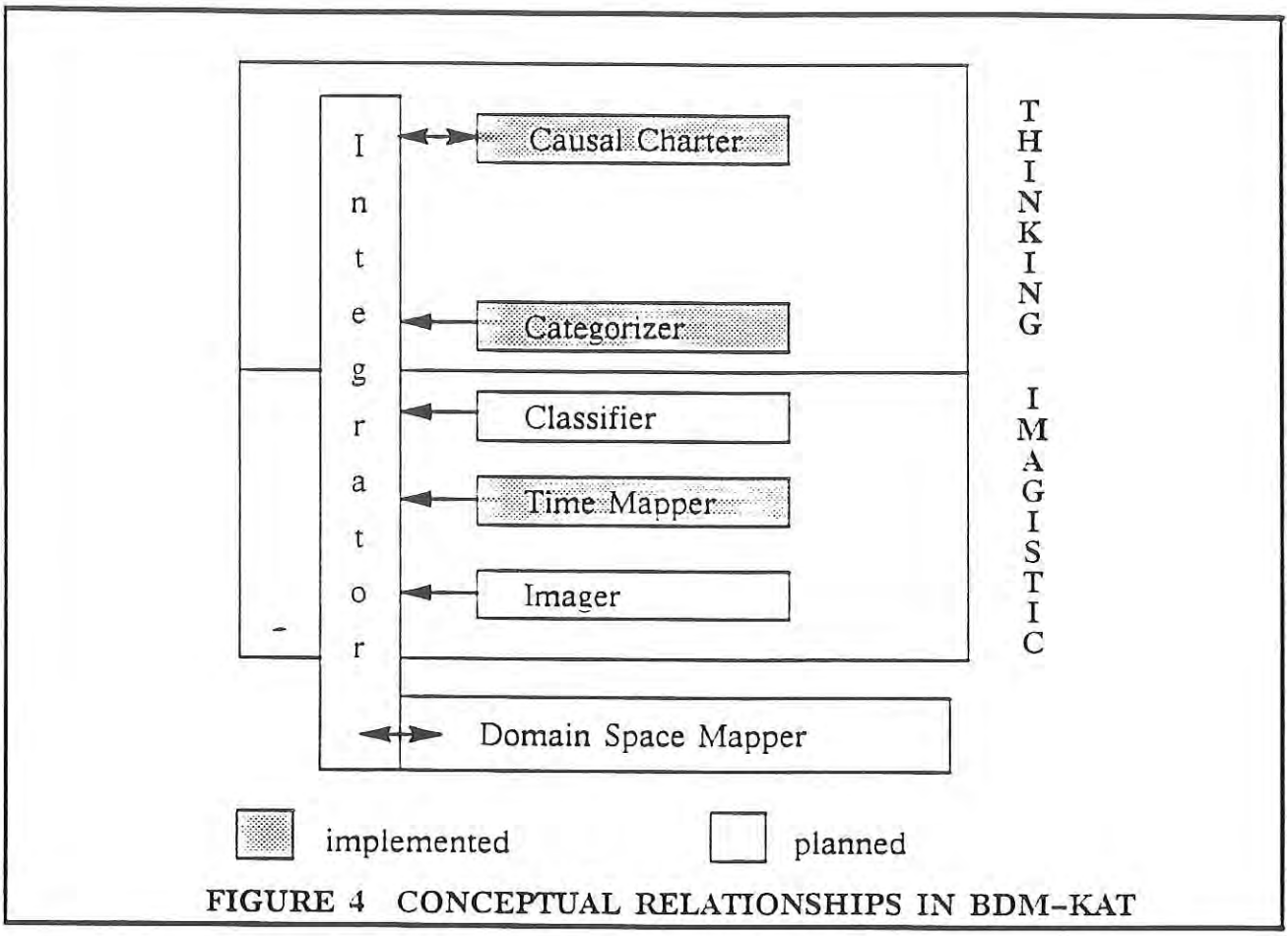


FIGURE 4 CONCEPTUAL RELATIONSHIPS IN BDM-KAT

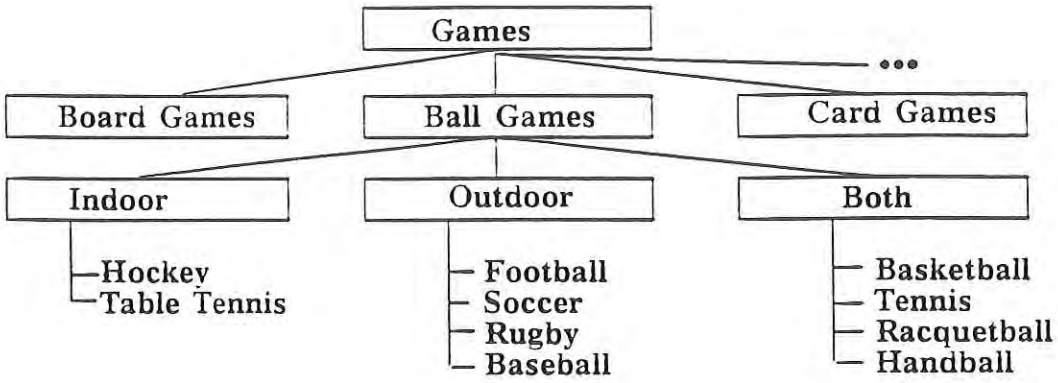


FIGURE 5 SIMPLE TAXONOMY OF GAMES

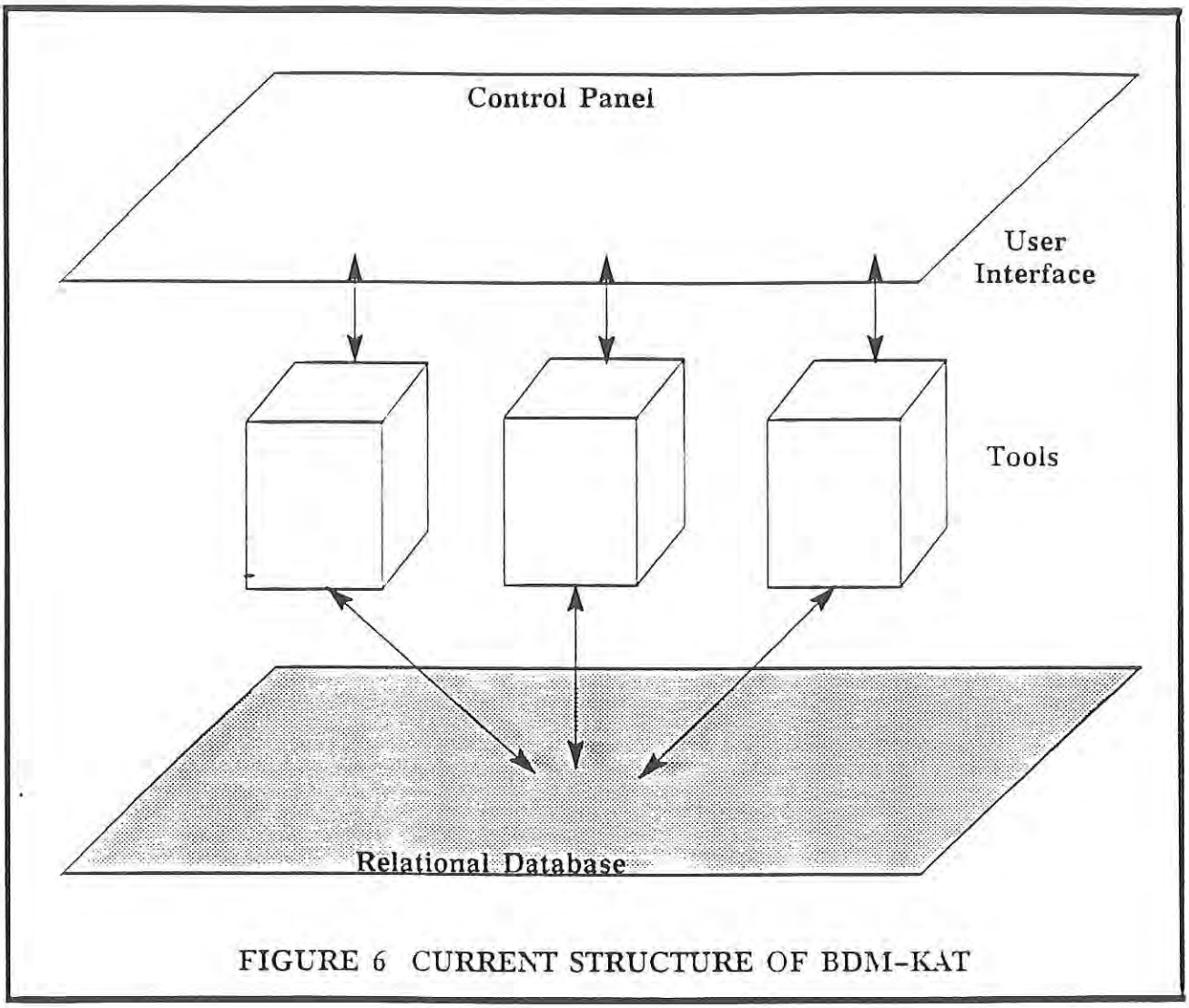


FIGURE 6 CURRENT STRUCTURE OF BDM-KAT

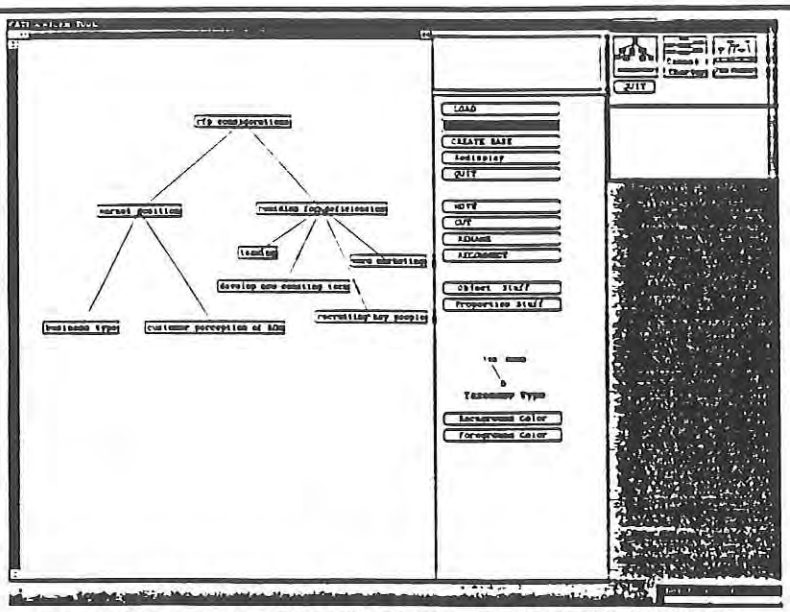


FIGURE 7 CATEGORIZER SCREEN DUMP

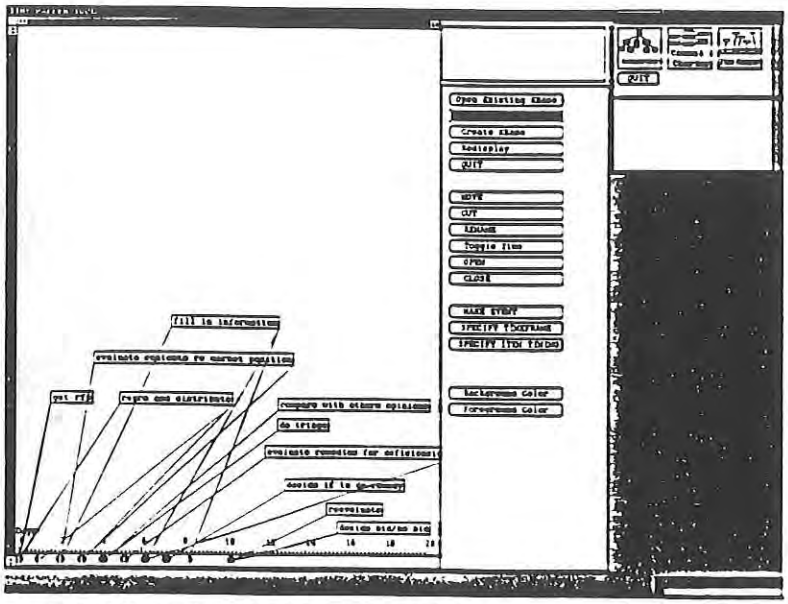


FIGURE 8 TIME MAPPER SCREEN DUMP

Rapid Prototyping of Intelligent Controller Knowlege Bases Using the BDM Knowledge Acquisition Toolkit (BDM-KAT)

Juliana S. Lancaster
Leo M. Vekker
Brian G. Kushner
BDM International, Inc.
and
Edward S. Manukian
University of Maryland

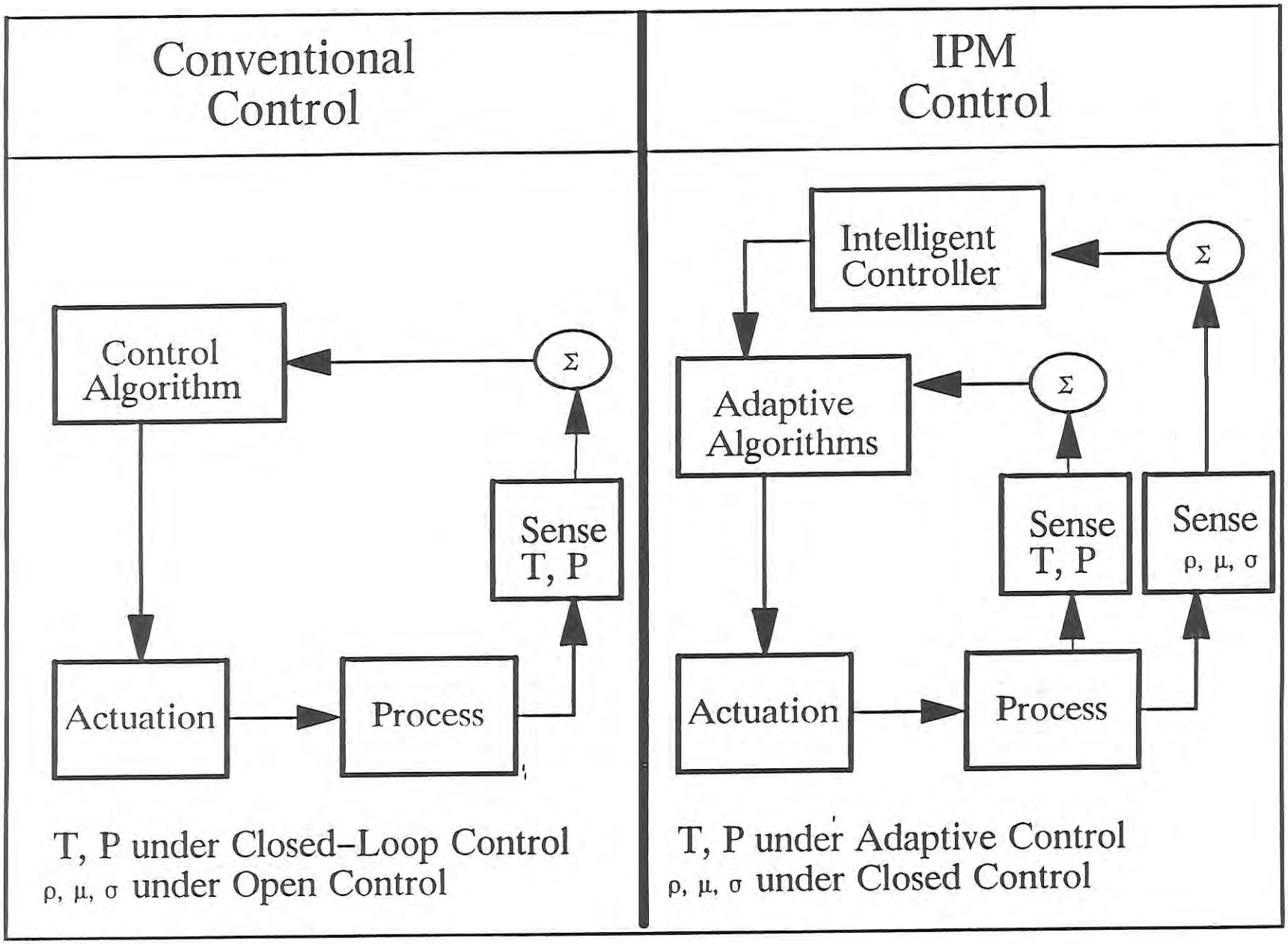
Abstract

The development of new systems, both commercial and military, has come under a serious, heightening handicap in recent years by a trend of lengthening time-spans in the technological developments of both materials and product. Demand for materials advances now face a lag of as much as 30 years between laboratory development of a new advanced material (e.g., P/M superalloys or high temperature epoxys) and its integration in operational components. The goals of an approach termed Intelligent Processing of Materials (IPM) are to make use of new potentials more quickly and to support improved control of advanced materials processes. IPM approaches support the development and introduction of sensors to measure microstructures and infer component properties which, coupled with the inherent flexibility of a knowledge-based (i.e., intelligent) controller, could provide the capability to tailor "specialty" materials and their processing for particular applications. Estimates indicate that IPM has the potential to reduce the transfer of technology from the laboratory to production from a time scale of multiple years to one measured in months. The end goal is to improve the quality, reliability, and yield of advanced materials and to accelerate their insertion in tomorrow's defense technologies.

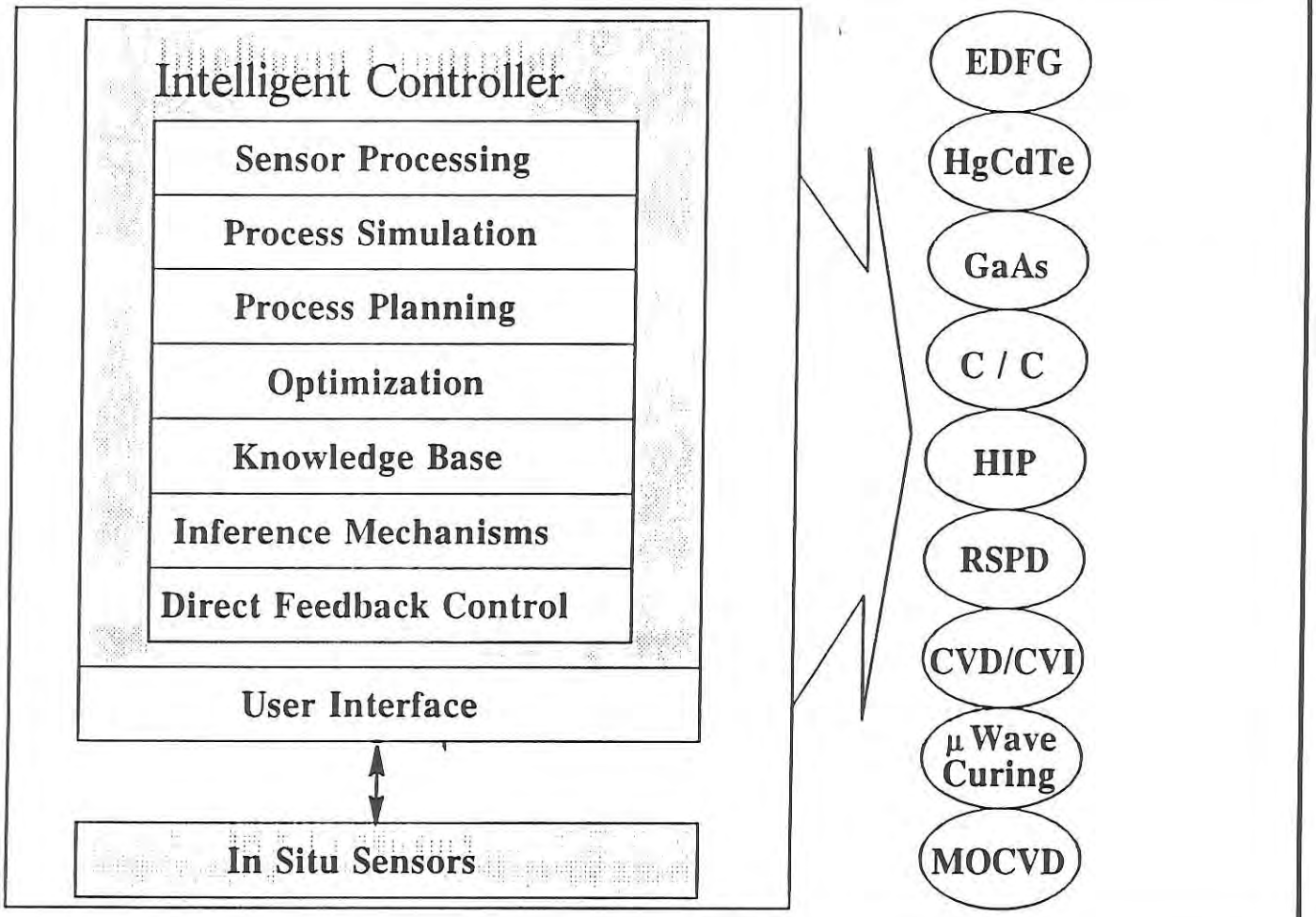
BDM has been addressing the requirements for intelligent control of Hot Isostatic Pressing (HIP) and the larger issue of effective and efficient knowledge engineering for IPM in related projects over the past three years. In these projects, working prototypes of both an intelligent controller and of a knowledge engineering toolkit (*BDM-KAT*) have been developed. This paper discusses intelligent control and its relationship to effective knowledge acquisition (KA) and reviews the rationale and theoretical support for *BDM-KAT*.

RAPID PROTOTYPING OF INTELLIGENT CONTROLLER KNOWLEDGE BASES USING THE BDM KNOWLEDGE ACQUISITION TOOLKIT

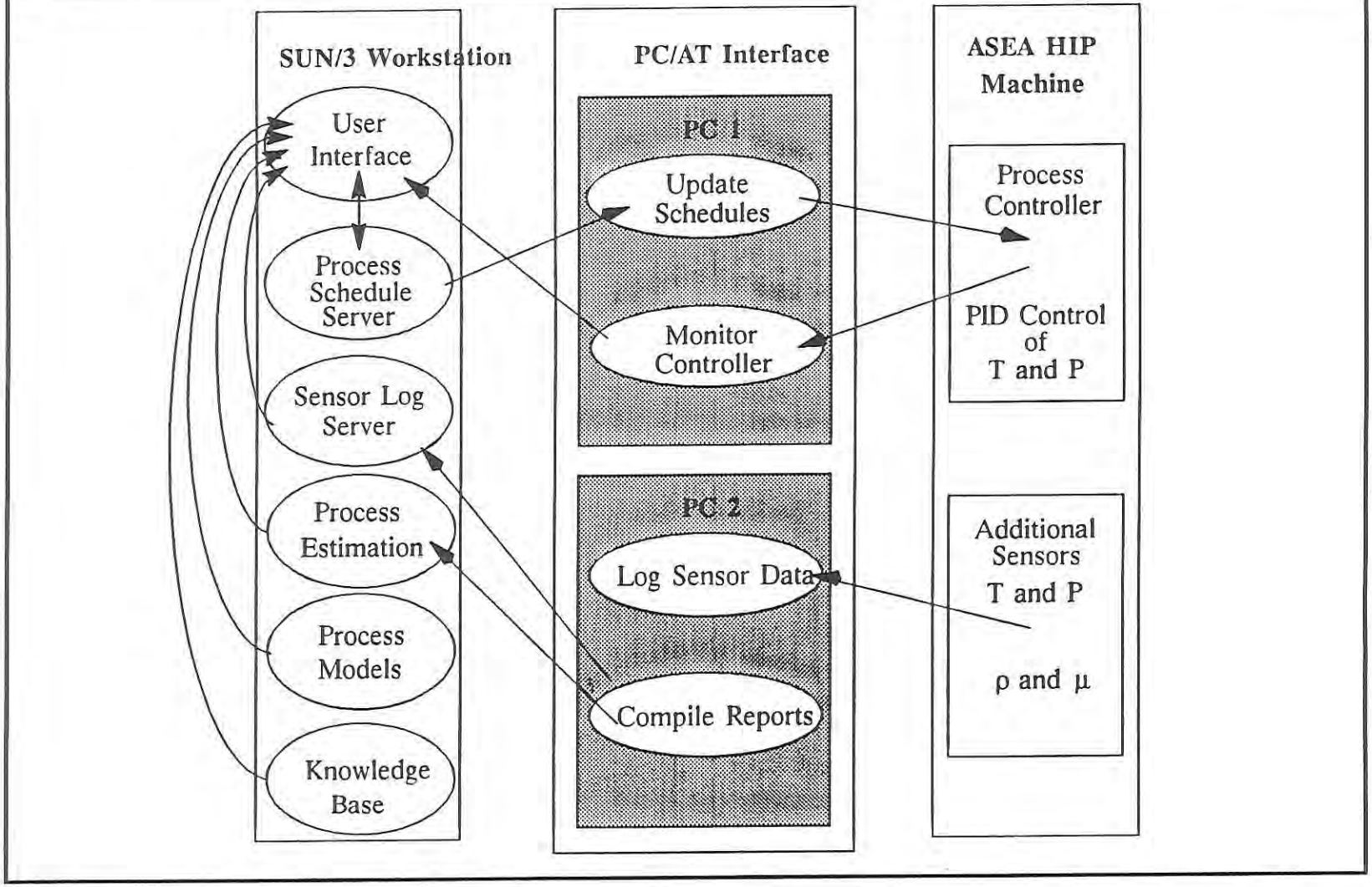
Juliana S. Lancaster
Brian G. Kushner
Leo M. Vekker
BDM International, Inc.
and
Edward S. Manukian
University of Maryland


Conventional Control
Control Algorithm
Σ
Sense T, P
Actuation
Process
T, P under Closed-Loop Control
ρ, μ, σ under Open Control
IPM Control
Intelligent Controller
Σ
Adaptive Algorithms
Σ
Sense T, P
Sense ρ, μ, σ
Actuation
Process
T, P under Adaptive Control
ρ, μ, σ under Closed Control

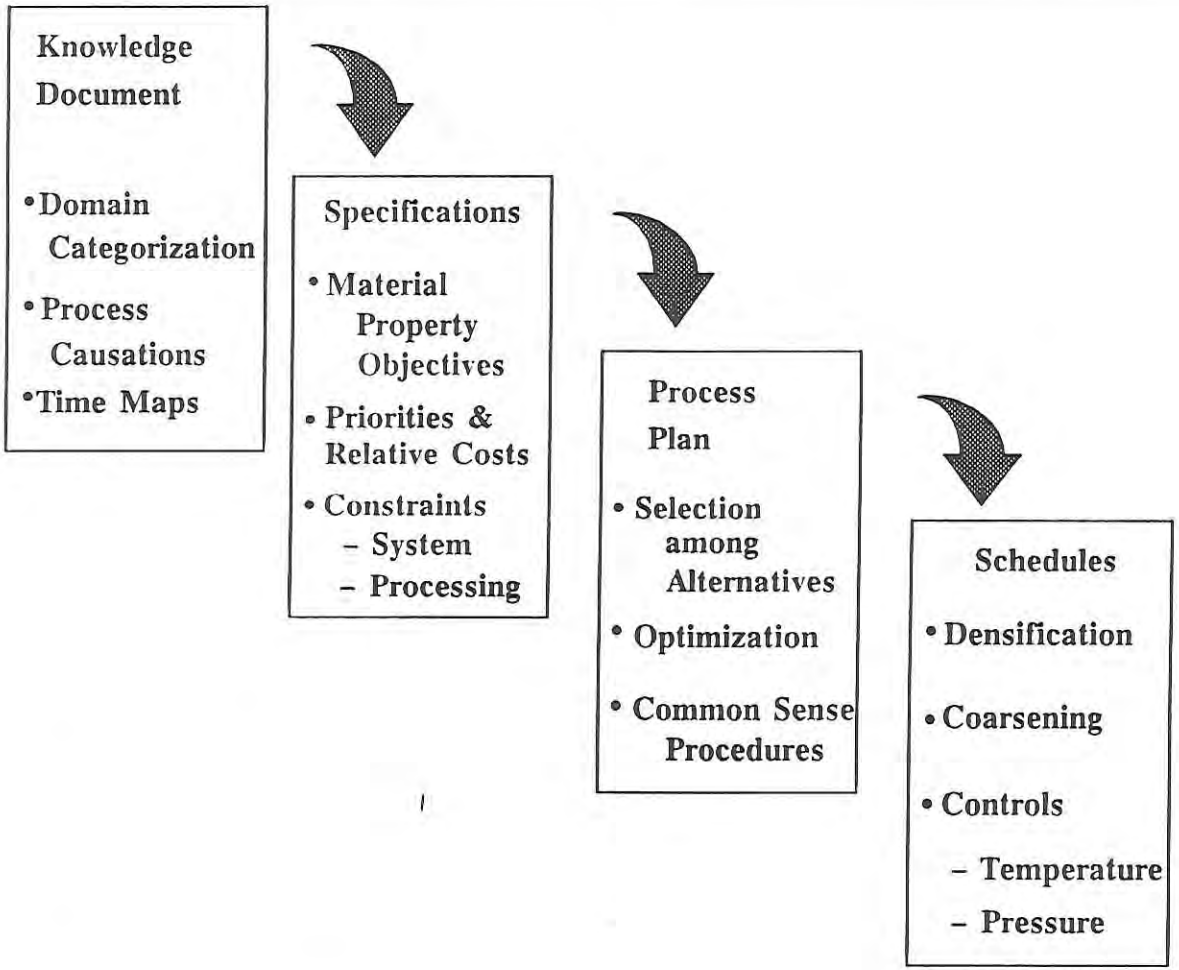
Components of an Intelligent Control System



BDM's HIP Controller Architecture



Knowledge Document Uses



HIP Process Specification

Interface of Knowledge Document with Computational Facilities

- **Material Property Objectives**
  - -- **Final density** ρ_{final}
  - -- **Final grain size** d_{final}
- **Priorities (Relative Costs)**
  - -- **time** w_{time}
  - -- **diffusion depth** w_{dd}
  - -- **grain growth** w_{gg}
  - -- **energy consumption** w_{energy}
- **HIP System Constraints**
  - **Maximum temperature** T_{Max}
  - **Maximum pressure** P_{Max}
  - **Maximum ramp rates** $\dot{T}_{Max}$, $\dot{P}_{Max}$
- **Processing Constraints**
  - -- **Avoid densification front** $\dot{\rho}_{Max}$, ∇T_{Max}
  - -- **Avoid canister buckling** ∇P_{Max}
  - -- **Avoid excessive shape distortion** $\dot{\rho}_{Max}$, ∇T_{Max}, ∇P_{Max}

Types of Knowledge Required for IPM Domain

Domain Structure

- **Knowledge of Materials Properties and Interrelationships**
- **Knowledge of Novel, Incompletely Characterized Materials**

Causal Networks

- **Process Control and Causality Knowledge**

Constraints

- **Temporal Knowledge of Process**
- **Knowledge of Where Process Models are Valid**

Uses of IPM Knowledge in Intelligent Control

Domain Structure

- complete specification sheet
- categorize faults and failures

Causal Networks

- material selection
- active control and diagnosis

Constraints

- schedule optimization
- priority ordering and schedule selection

Desired Features of a Knowledge Acquisition System

A Knowledge Acquisition System should:

- reduce the level of burden on the domain expert
- reduce the dependencies on individual knowledge engineer skills
- reduce the requirements for translation to KRL
- reduce the domain dependence of tools
- provide a controlling structure for a KA effort
- provide a mechanism for selecting appropriate tools/techniques

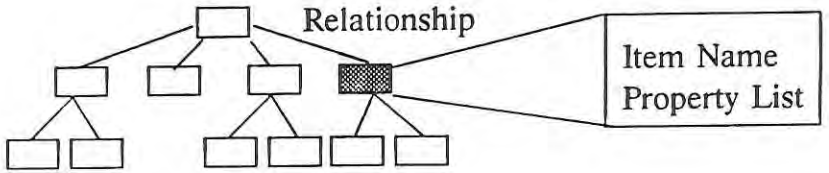
Our Solution was to Use a Cognitive Basis for KA

Categorizer

PURPOSE: Capture stable, underlying domain structure

METHOD: Elicits class/subclass and part/whole hierarchies
Tracks relationships and object properties
Differentiates necessary and contingent properties

INTERFACE: Graphic representation of developing taxonomy

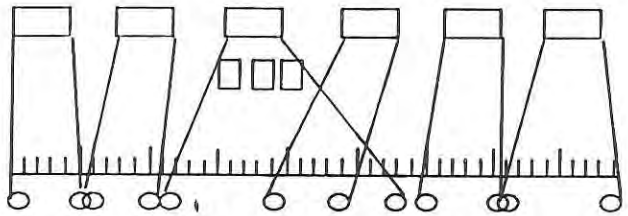


Time Mapper

PURPOSE: Decompose process into event hierarchy
Capture temporal relationships

METHOD: Elicits event/subevent structures
Elicits temporal unit and duration for events
Elicits start point and min/max durations for events

INTERFACE: Graphic representation of event sequence and timing



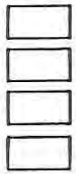
Causal Charter

PURPOSE: Elicit cause–effect chains in processes and events
Provide verification of knowledge elicited with other tools

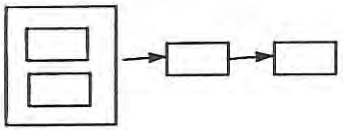
METHOD: Elicits functions, structural parts and causal links
Can elicit higher order "rules"

INTERFACE: Textual representation of objects, functions, and structures
Graphic representation of causal chains

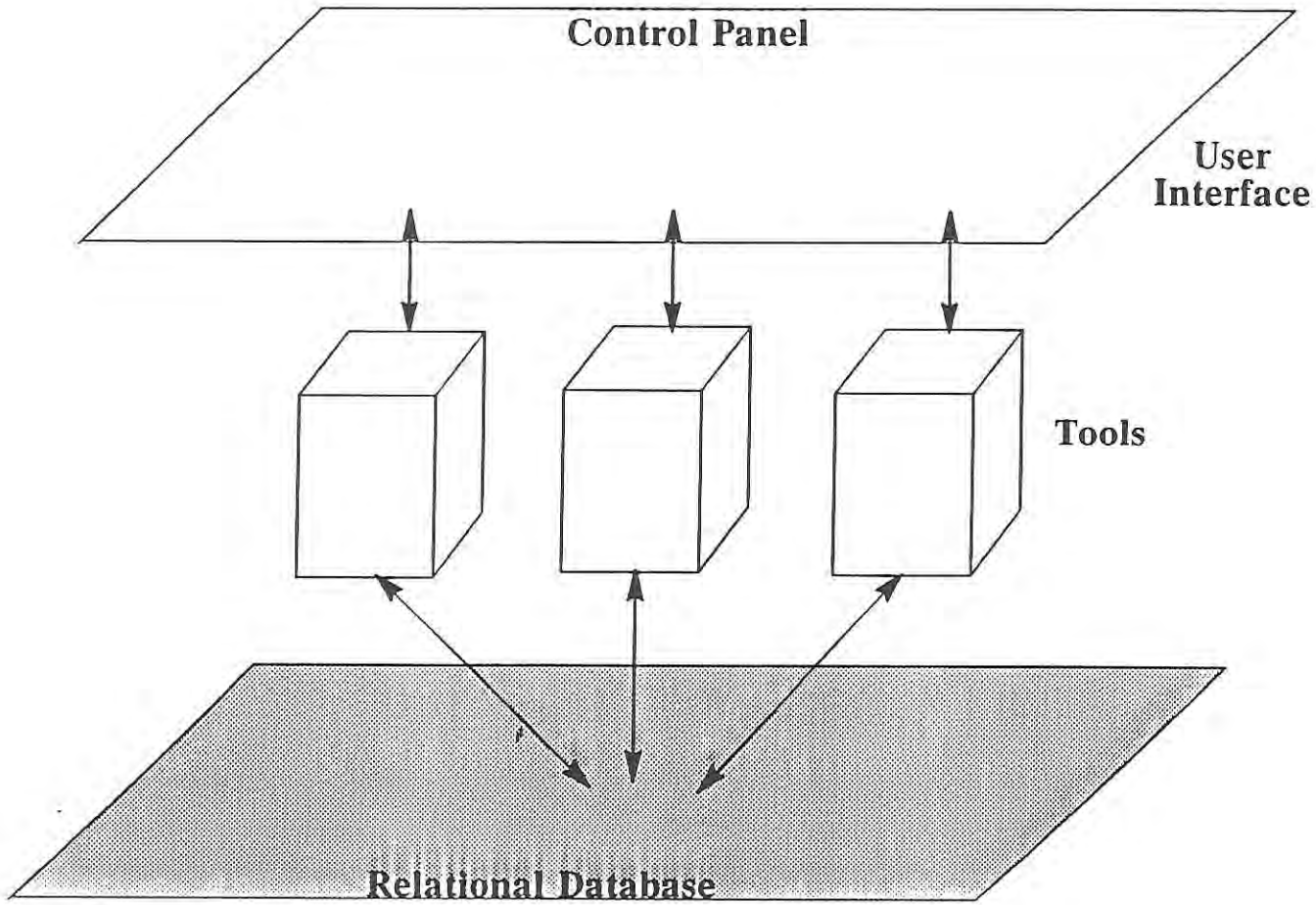
Item List



Causal Chain



Structure of BDM-KAT



A Cognitively Valid Knowledge Acquisition Tool

Juliana S. Lancaster
Christopher R. Westphal
Karen L. McGraw
The BDM Corporation
Advanced Technology Division
1300 N. 17th Street, Suite 950
Arlington, VA 22209

Introduction

As more (and more) knowledge-based systems are designed for operational use, it becomes apparent that the knowledge engineering (KE) process of transferring domain expertise into the system consumes the single largest block of development time (Feigenbaum, 1977; Hayes-Roth, Waterman, & Lenat, 1983). To reduce the time required for this stage, numerous tools have been developed that automate or systematize the elicitation and transfer of knowledge from expert to knowledge base. Among these advances are developments in machine learning that enable an expert system to acquire its knowledge from "experience" (Michalski, 1980; Michie, 1982), expert system shells that allow domain experts to write their own rules (e.g., MacSMARTS), and knowledge engineering tools that assist a knowledge engineer in his/her work. The material reported here describes BDM's development of a computer-based set of tools, known collectively as BDM-KAT.

BDM-KAT has been designed and implemented to aid a trained knowledge engineer in eliciting expert knowledge about well-structured domains. BDM-KAT is based on principles of human cognition, particularly those relating to the representation and structure of knowledge in memory. BDM-KAT provides an environment in which the knowledge engineer and domain expert discuss, manipulate, and record the objects and relationships that provide the domain with its structure and predictability. As the elicitation progresses, a database of basic domain components is developed. Using this database, a knowledge base and inferencing structure can be built to operate in the selected domain. It is important to note that BDM-KAT *can* elicit rules about the domain and its operation, but that is is not *required* to do so.

Most knowledge engineering tools force domain experts to present their knowledge in the format on which the tool is based (e.g., repertory grids, fuzzy sets). This requirement may force the experts to translate the expression of their knowledge from a familiar, useful form to a novel, untested one. BDM-KAT avoids this restriction through the use of recognized conceptual mechanisms such as categorical structures (Rosche, Mervis, Gray, Johnson, & Boyes-Braem, 1975), causal models, and timelines of events (Barsalou, 1986). In developing this framework, several characteristics of human knowledge organization and reasoning surfaced that are critical to the KE process: the structure of knowledge, the stability of knowledge, and the representation of knowledge. Each is briefly discussed below.

Structure

Categorical Structure of Knowledge: People tend to divide the world into relatively neat categorical structures. While for laymen the categorical divisions are frequently unstable (e.g., individual items may change their positions in terms of both category membership and typicality within a category (Barsalou, 1986)), experts in a specific field hold stable and shared categorizations of their domain (Rosche, 1975). A substantial portion of one's understanding of a domain rests in knowledge about the items that appear in the domain, their individual and collective characteristics, and the relationships between them. Because this organization is natural to the domain expert, knowledge elicited within this structure is likely to be more easily obtained and more accurate.

Temporal Organization of Events: In addition to dividing objects in the world into categorical sets, people structure their knowledge and memories of events into orderly sequences. This knowledge appears to be organized internally into "mental time line models" in which the temporal relationships between episodes are specified (Barsalou, 1986). Where the episodes or process events can be decomposed to enabling or sub-events, hierarchical relationships exist between them similar to those between levels in a categorical taxonomy. This structural nature of event memory can be exploited in eliciting knowledge about processes that occur over time. Drawing from this model the domain expert can tap and share the order, temporal characteristics and relationships, and hierarchical associations between steps in the process.

Causal Structures in Knowledge: Finally, experts can be partly defined as those individuals who understand the *why* and *how* of a particular domain. All people, even novices, construct causal mental models of the world around them (Kelly, 1955). However, causal models held by a novice may reflect incorrect assumptions or focus (e.g.,

correlations), while those held by an expert tend to be more accurate. As with categorical and temporal organizations of knowledge, causal models can be elicited directly from an expert by focusing interactions on the expert's explanations for states and conditions in the domain of interest.

Of these three types of knowledge, causal models are the most similar to the rules usually sought in knowledge acquisition. The categorical and temporal features of a domain are also critical for understanding, reasoning, and problem solving in the domain, but are difficult to elicit manually in the forms needed for knowledge-base system development. BDM's research efforts in the design of the Knowledge Acquisition Tool (KAT) have been oriented toward developing a method and means for eliciting and utilizing expert knowledge in a form compatible with the expert's internal model.

Stability

The design philosophy behind BDM-KAT is based on an application of the theories of Vekker* (1974, 1976, 1981). A critical component of his theory states that expert knowledge about a domain is stable across time and perspective. For the purposes of knowledge elicitation, the stability of expert knowledge has two implications. First, knowledge elicited in one session can be compared to the results of other sessions for validation purposes, thus providing an ongoing capacity for checking the accuracy and completeness of the growing knowledge base. Second, the "expertness" of an individual can be determined by monitoring the stability of his/her knowledge across sessions. To the extent that stability is not maintained in a segment of the domain, the knowledge engineer may decide to pursue efforts in that sub-domain with a different domain expert.

Modality

Research has indicated that human knowledge is represented in both visual and verbal modalities (Vekker, 1981; Anderson, 1984; Paivio 1986). Vekker supports the theory that object names and properties are verbally represented while spatial, logical, and temporal relationships are more likely to be visually represented in a person's cognitive model. This perspective on human knowledge representation has important implications for the design and structure of user interface representation and internal human representation that will enable more efficient interactions between user and system.

BDM-KAT Design

BDM-KAT has been designed to enable a knowledge engineer to elicit knowledge based on both the structure and modality in which it is most likely to be represented in the domain expert's internal cognitive model. To accomplish this, BDM-KAT consists of three cooperating knowledge engineering tools: Categorizer, Time Mapper, and Causal Charter. *Categorizer* elicits knowledge about the stable, underlying structure of a scientific domain via the drawing and filling in of a tree-like taxonomic structure. *Time Mapper* provides a context in which the domain experts position items on time lines to express knowledge about the temporal characteristics of and relationships between events and subevents within a process. Using *Causal Charter* the knowledge engineer elicits cause-effect knowledge for domain functions and events. Each tool provides the knowledge engineer and domain expert with a means for 1) focusing attention on the aspect of the domain being elicited, 2) recording information as it is elicited, and 3) verifying the information as it is elicited.

Each tool uses both textual and graphic interface representations. All tools output to and interact with a single database, so knowledge elicited by any tool can be used in future elicitation sessions with other tools. The use of such a database also provides an excellent knowledge browsing facility and content for the development of knowledge documents.

Categorizer

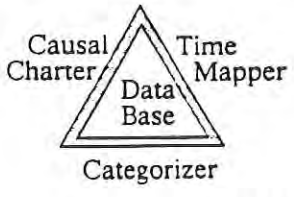
Categorizer elicits the underlying structure of a domain by requiring the expert to construct a taxonomy or set of taxonomies. As discussed earlier, categorical knowledge is commonly held for any domain or topic area. Furthermore, it is critical for constructing a domain model because the properties of specific objects and object classes will dictate much about how the knowledge can be used. In Categorizer, the expert initially creates and names objects that exist in the taxonomy and places them at the appropriate position within the structure. The expert works primarily in a top-down fashion, but the tool does not restrict the level at which objects can be named. After the taxonomy has been developed the essential, contingent, and second-order properties of each domain object are elicited from the expert. A clear knowledge of object properties is necessary for the success of the overall knowledge engineering effort because it will be these properties that will determine how the model can be perturbed during future verification activities. Object properties are best elicited in tandem with object taxonomies in part because they follow from the structure of the taxonomy and also because, while the expert's attention is focused on that taxonomy, s/he will be less likely to make errors.

Time Mapper

As Categorizer elicits taxonomic and part-whole relationships among object, Time Mapper elicits the structure of events or actions that occur during the execution of a process. Specifically, Time Mapper elicits the hierarchical relationships among events and subevents (at multiple levels), the overall duration of each event, and the specific starting and ending times of each subevent within an event. This elicitation effort is critical to developing a full process model because it provides both the decomposition of the process into well specified components and the temporal relationships necessary for control. The expert is asked to name the process as a single event and then to name the subordinates of that event. This cycle iterates with increasing specificity such that the subevents at one level are considered as events at the next, until the events can be decomposed no further.

Causal Charter

The third tool, Causal Charter, is used to elicit high level "rules" and low level cause-effect chains. The rules are intrinsically implicit and are supplied by the expert through a description of an item to be "charted". This item under consideration may be either a function of a domain object (specified during Categorizer as a functional property of that object) or a subevent from the process decomposition state of Time Mapper. The structural elements in the item description are separated from the additional functions specified and are grouped together to indicate those elements that support each function in the description. Thus, the generation of clusters of structural elements (as units) are related to specific functions. The expert is asked to name those items from the description that provide support for each function given in the description. When specifying the cause-effect chains the elements are selected from the mechanism in the order in which they are caused or required by the mechanism. Thus a completed chain indicates the process by which the final state or element is reached. Causal Charter can also serve as a verification mechanism on some components of the categorical and process knowledge elicited previously.



*Professor Lev M. Vekker is the former head of Cognitive Psychology at the University of Leningrad. Special thanks to Roger Geesey, Carl Friedlander, and Larry Reeker. Work described in this paper was supported by the U.S. Defense Advanced Research Projects Agency, through the Air Force Office of Scientific Research under contract number F49620-86-0036 and the Army Research Office under contract number DAAL03-87-C-0019. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing the official policies, either expressed of implied of the Defense Research Advanced Research Projects Agency or the U.S. Government.

References

Anderson, J.R. (1984). *Representational types: A tricode proposal*, Technical Report #82-1, Office of Naval Research, Washington, D.C.

Barsalou, L.W. (1986). The content and organization of autobiographical memories (unpublished draft).

Feigenbaum, E. (1977). The art of AI. *Proceedings of the Fifth International Conference on Artificial Intelligence.*

Hayes-Roth, F., D. Waterman, & D. Lenat (1983). *Building Expert Systems* Addison-Wesley.

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

Michalski, R. (1980). Knowledge engineering through conceptual clustering: A theoretical framework and algorithm for partitioning data into conjunctive concepts. *International Journal of Policy Analysis and Information Systems, 4(3)*, 219-243.

Michie, D. (1982). The state of the art in machine learning. In D. Michie (ed) *Introductory Readings in Expert Systems.*

Paivio, A. (1986). *Mental Representations: A Dual Coding Approach*, New York, Oxford University Press.

Rosche, E., C.B. Mervis, W.D. Gray, D.M. Johnson, & P. Boyes-Braem (1975). Basic objects in natural categories. *Cognitive Psychology, 8*, 382-439.

Vekker, L.M. (1981) *Mental Processes: The Agent, the Experience, Action and Consciousness* Vol. 3, Leningrad University Press. (In Russian).

Vekker, L.M. (1976) *Mental Processes: Thinking and the Intellect.* Vol. 2, Leningrad University Press. (In Russian).

Vekker, L.M. (1974) *Mental Processes.* Vol. 1, Leningrad University Press. (In Russian).

KNOWLEDGE-BASED SYSTEMS APPLIED TO ADVANCED MATERIALS MANUFACTURING

Andrew B. Levy and John J. Wlassich

The BDM Corporation
1300 North 17th Street
Suite 950
Arlington, Virginia 22209
(703) 247-0340

ABSTRACT

The U.S. Department of Defense (DoD) is planning a new generation of weapons systems that require the production of advanced materials in quantities that are cost-prohibitive, given current manufacturing techniques. This paper introduces the Defense Advanced Research Projects Agency, Defense Sciences Office (DARPA/DSO) Intelligent Processing of Materials (IPM) program and describes its projected impact on the reduction in cost of advanced materials manufacturing through the utilization of knowledge-based systems and intelligent control.

1.0 INTRODUCTION

Historically, weapons systems in the U.S. have been designed to fulfill multiple roles. Examples are the F-111 fighter/bomber and the Armored Personnel Carrier that serves as a command and communications vehicle. The motivation for multiple role weapons is one of force multiplication; procuring fewer weapons while maintaining effective readiness. Versatile weapons imply a degree of innovation and sophistication over single function arms. Often, advanced technology systems demand modes of operation that exceed current materials properties. The new materials must withstand higher temperatures, greater stresses, and harsher environments than their predecessors.

Expert Systems and Intelligent Manufacturing

The U.S. Department of Defense (DoD) has begun plans for a new generation of weapons systems that require the production of advanced materials in quantities that are cost-prohibitive, given current manufacturing techniques. An examination of the state of advanced materials manufacturing reveals the primary factors that contribute to the high cost. The manufacturing techniques for advanced materials are in a stage that is not far removed from initial laboratory-scale processing. Fundamental insight into the process kinetics is usually poor or nonexistent. The same can be said of sensors of process state variables. Processes are, in general, controlled by heuristics that are compiled from numerous runs and sensitivity testing. This trial-and-error methodology leads to predictably low levels of production yields -- outright losses are high, typically greater than 50 percent -- thus making the manufacturing process appear more as an art than a technology. Furthermore, material uniformity is poor and reproducibility is low. Fabrication cycle times are conservative so as to ensure a comfortable margin of safety that as least a portion of the product will meet specifications. Experimentation to scale up to larger production quantities is often inhibited by the high cost of failures.

The Defense Sciences Office of the Defense Advanced Research Projects Agency (DARPA/DSO) has responded to the need for innovative, economical production techniques for advanced materials critical to DoD mission requirements through the formulation of the Intelligent Processing of Materials (IPM) program. IPM targets GaAs, titanium aluminide intermetallics, and carbon/carbon composites as prime emerging materials that could benefit from a pointed research thrust in order to achieve economical production means. The challenge is to accelerate the transition from materials processing in a laboratory environment to fabrication techniques in a production environment. Today, a complete cycle of research, development, and application requires about 15 years. The mission of the IPM program is to reduce this cycle by half for specific materials in order to meet DoD's timetable for deployment of new systems, such as the National Aerospace Plane that is projected for 1993. The tools proposed by IPM for accelerating the transition are analysis and synthesis of the materials process kinetics to form models of the state metastasis, in conjunction with *in-situ* sensors to monitor actual states. The vehicle for the accelerated transition of laboratory research to yield enhanced manufacturing techniques is knowledge-based systems.

Figure 1 depicts an approximate relationship between models, sensors, and heuristics in advanced materials production. The focus of IPM is to accelerate the movements of materials processing techniques from a position on the left to a more economical position in the right. Phase I of IPM encompasses research in addition to capitalization on the gains afforded by developing new sensors and capturing process heuristics in knowledge-based systems.

One might ask, if the models and sensors are part of the IPM research effort, how can one instantiate heuristics given that experts will not exist who have used the new models and sensors? IPM overcomes this problem by structuring the program to meld existing heuristics, data bases, models, and sensor information into a computer-based workstation from which the processes can be controlled through human intervention. This workstation approach is illustrated in Figure 2. By the end of Phase I, the new expertise will be captured in knowledge-based systems in order to emulate and enhance human control actions, forming an intelligent controller for use in Phase II of IPM. Phase II will tackle scaling and implementation issues to apply the suite of analytical models, *in-situ* sensors, and knowledge-based systems to a manufacturing environment. To promulgate the transfer of this innovative manufacturing approach for the benefit of American industry, a microfactory will be built that embodies the essential elements of intelligent control of the advanced material at a production level, for showcase demonstrations, and actual material production in bulk quantities.

The remainder of this paper will focus on knowledge based systems applied to manufacturing problems and the approach taken by The BDM Corporation in the area of knowledge acquisition. In addition, a summary of one IPM program, Hot Isostatic Pressing, will be presented as an example of the implementation issues addressed by IPM.

2.0 IPM AND EXPERT SYSTEMS

The field of expert systems is a discipline within the applied artificial intelligence field wherein human expertise in a specialized domain is codified and implemented on a computer. This coded expertise may then be coupled with other processes, such as an inference engine, a user interface, and a report generator module, in order to build an integrated application program that uses the coded knowledge. One of the benefits that may be attained from expert systems is that, in some cases, high levels of performance are achievable with computerized expert systems in a fraction of the time that would be required for human experts to receive the necessary training and education to accomplish the same task.

Expert systems have received a great deal of attention recently in professional and academic circles. Much of the attention, however, is due to the relative successes of a small number of expert systems being used for research purposes or in professional fields. And although there remains no unanimity of the definition of what an experts system is, such systems typically possess a number of characteristics that provide a clear distinction. Expert systems are heuristic by nature, using "rules of thumb" to search for the best solution in a space of

solutions in which there is not necessarily a single "right solution." The type of information processing performed by an expert system is heavily inferential.

For this reason, expert systems tend to address problems that do not have conventional algorithmic solutions; and, if the expertise that is captured in the program is deep enough, they come to much the same conclusions as do human experts. Conventional programs, on the other hand, are algorithmic processes that execute procedural instructions, usually in a repetitive fashion. There is no reason, of course, that expert systems cannot incorporate algorithmic processes where they are known and useful. We know that human experts do the same, with pencil and paper or a computer.

One of the critical problems that has arisen relative to the development of expert systems and their component knowledge bases deals with the process by which knowledge is acquired. In many cases, knowledge acquisition is a more formidable problem than locating the expertise or operating the expert system development software on a computer. In fact, this problem is often labeled the "knowledge acquisition bottleneck." Simply stated, the knowledge acquisition bottleneck refers to the problem of extracting domain knowledge in a structured fashion from one or more experts and representing it in a logical form that can be processed by the modules of a expert system. Knowledge acquisition is inherently time intensive and involves a number of data transfers between the expert, the knowledge engineer, and the computer, that can contribute to a loss of knowledge integrity. Building initial knowledge bases for a wide variety of domains, therefore, remains one of the largest problems in development and application of expert systems today. Let's examine in greater detail why knowledge acquisition is a difficult problem.

Typical approaches to building knowledge bases for expert system use rely on interactions between a domain expert and a programmer. In this manner, the expert's knowledge is communicated to the programmer, who may impose a new interpretation or organization onto the information before encoding it into the system. For those applications in which expertise is to be extracted from multiple knowledge sources, no mechanism exists for resolving any conflicts that may arise. In summary, typical approaches to knowledge acquisition often compromise the integrity of the gathered data.

It is this knowledge acquisition bottleneck that has been identified by DARPA as the critical problem in transitioning knowledge relating to strategic systems, such as the knowledge of advanced materials manufacturing, from the laboratory to the factory. The IPM program seeks to reduce the time required for transitioning the associated processing techniques into full manufacturing environments. Doing so will involve the expertise of laboratory materials scientist and process engineers as well as the use of expert systems, *in–situ* sensors, and process models. The

integration of these systems will facilitate greater control over the materials processing environment and will reduce the long lead-time between initial materials development and eventual materials production and utilization.

2.1 A Knowledge Acquisition Tool

Because of the specialized nature of some of the subsidiary materials processing fields, as well as the small number of industry experts in those fields, the IPM program will employ expert system technology to develop an intelligent knowledge acquisition system. This portion of the IPM program, known as the Knowledge Acquisition Tool (KAT) project, is aimed at making the most efficient and effective use of the limited number of domain experts in some of the materials processing fields that have the highest potential for payoff in DoD systems. KAT is envisioned to be an intelligent supplement to the knowledge engineering process because it is being designed to provide structured mechanisms that can be used by the knowledge engineer for eliciting information about materials processing.

It should be stated that KAT is not intended to replace the knowledge engineer or the knowledge engineering process. It is not intended to be a system that builds materials processing expert systems. Rather, it is an intelligent tool that assists a knowledge engineer in the construction of a knowledge base. To that degree, KAT is designed with interchangeable domain-specific libraries and models. It therefore can be used to construct knowledge bases in virtually any domain for which a library of knowledge objects may be defined. KAT uses graphics-based drawing windows and multiple representations of knowledge objects to bring the domain expert closer to the maturing knowledge base and to allow the domain expert to freely explain a process, model, or event. The use of such a tool in itself constitutes a major advancement in the knowledge engineering process.

Using KAT, the knowledge engineer and domain expert will iterate through three querying modes in which the expert will be prompted to answer different types of questions about a particular materials processing domain. In the first mode, known as the clarification mode, the expert will be provided with facilities for creating an AND/OR graphical representation of the process. Once a certain amount of information is encoded in this format, the users will switch into a second mode called the prediction mode. At the start of the prediction mode, the knowledge structure created during the clarification mode will be converted into a flowchart display. The users will be prompted to answer a series of questions about each subprocess represented in the defined structure. For example, the domain expert might be asked to estimate ranges for temperature, pressure, and exhaust gas ratios at each point identified thus far in the process. In addition, the expert will be asked to make predictions about what could go wrong at each given point and to

provide suggestions for redesign that might lead to avoidance of identified problems. Once these queries are completed, the knowledge engineer will return to the clarification mode and will recycle through the questioning, thus adding to the knowledge base in an efficient, iterative manner.

Once the expert has iterated through the first two modes several times, a fairly detailed knowledge structure will be in place. At this point, the users will enter a third mode, termed the diagnosis mode. In this mode, the KAT facilities will provide a step-through presentation of the materials process as specified up to this point. A sensor panel will be displayed so that the users can see how certain process indicators change dynamically during the process. In addition, the users will be able to trace the pattern of the rule firings that effect the changes indicated by the sensors. In this mode, the domain expert will have opportunities to diagnose the process knowledge in the knowledge base created thus far by adding, changing, or deleting elements as necessary.

2.2 KAT Progress to Date

KAT is being designed and built at BDM on a Symbolics 3670 Lisp machine using the KEE™ and SimKit™ development environments.[1] KAT makes extensive use of the graphical and user interface functions within KEE as well as the design models, library representations of objects, clock, calendar, and various mathematical tools in SimKit. The event-driven nature of the SimKit simulation environment makes it ideal for representing materials processes as networks of objects and relations.

Even though the KAT contract portion of the IPM program has only been in place for several months, a substantial amount of design and programming has already been accomplished. Much of the design work for the clarification and prediction modes for the "Version 0.0" prototype of KAT has been completed and design work for the diagnosis mode is under way. Additionally, functional modules of the clarification and prediction modes have been coded and are undergoing enhancement and testing.

3.0 HOT ISOSTATIC PRESSING: AN IDEAL IPM PROGRAM

Hot Isostatic Pressing (HIP) is a process by which metal powders are compacted to nearly 100% of complete density in forms that approximate the net desired shape of a component. The primary benefit of HIP is the ability to produce parts made of metastable-phase intermetallics. Quenching ingots of titanium aluminide intermetallics does not transfer heat quickly enough to reach metastable phases. By rapidly solidifying atomized, molten titanium aluminide intermetallics to achieve

[1] KEE and SimKit are trademarks of IntelliCorp.

the desired phase in powder form, then compacting the solidified powder, one can yield bulk quantities of the required material.

A typical HIP cycle lasts three hours, during which the component is subjected to inert gas pressures of 2000 atmospheres and temperatures of 1000 degrees centigrade. The temperature and pressure profile during the cycle forms an open-loop schedule with the intent to achieve 100% densification. Figure 3 illustrates a process schedule. Presently, schedules are designed by relying on subjective rules compiled from numerous past experiments and production runs. Companies that manufacture materials via HIP have closely guarded scheduling heuristics resident at their facilities. Reduction in cycle times should focus on the excess time that is added to HIP schedules as a rough estimate to ensure 100% dense components. If monitoring of the actual component's density as a function of time were possible, cycle times could be optimized. Additional control to improve uniformity and cut outright losses depend on models of the yielding, creep, diffusion, and chemical bonds in the material as a function of the temperature and pressure schedule comprising a HIP run.

Phase I of IPM/HIP addresses the issues of increasing production yields and controlling microstructure by structuring a program that includes modeling of the process kinetics, capitalizing on unique density sensor, and instantly processing heuristics into an Intelligent Control approach. The laboratories of the Materials Science Division of the National Bureau of Standards (NBS) has been chosen for this program. An ASEA HIP chamber is ready for experimentation in addition to extensive facilities for test and evaluation of pressed samples. Models of the mechanical and chemical processes which occur during a HIP cycle exist for lead and copper. IPM/HIP targets extending this analytical understanding of titanium aluminide.

A new sensor based on eddy current measurements developed at the National Bureau of Standards (NBS) has been proven viable as an *in-situ* density sensor for hot isostatic pressing. Figure 3 illustrates the ASEA HIP chamber with the density sensor. HIP schedule expertise is resident and available for development and enhancement at NBS. In addition to HIP experts at NBS, a consortium of American manufacturers using HIP has been formed as an advisory council. Input from the consortium will be vital to the completion of successful research into HIP failure modes. These modes include densification fronts (premature densification of the outer surface of a component which inhibits further densification of the inside volume), component buckling, metal powder container leakage (which halts densification and causes oxides to form on titanium aluminide), and dimensional constraints on the metal powder container to avoid undesirable shrinkage patterns. Already, guidelines from the consortium members has yielded research direction valuable to the NBS scientists in the area of shrinkage modeling.

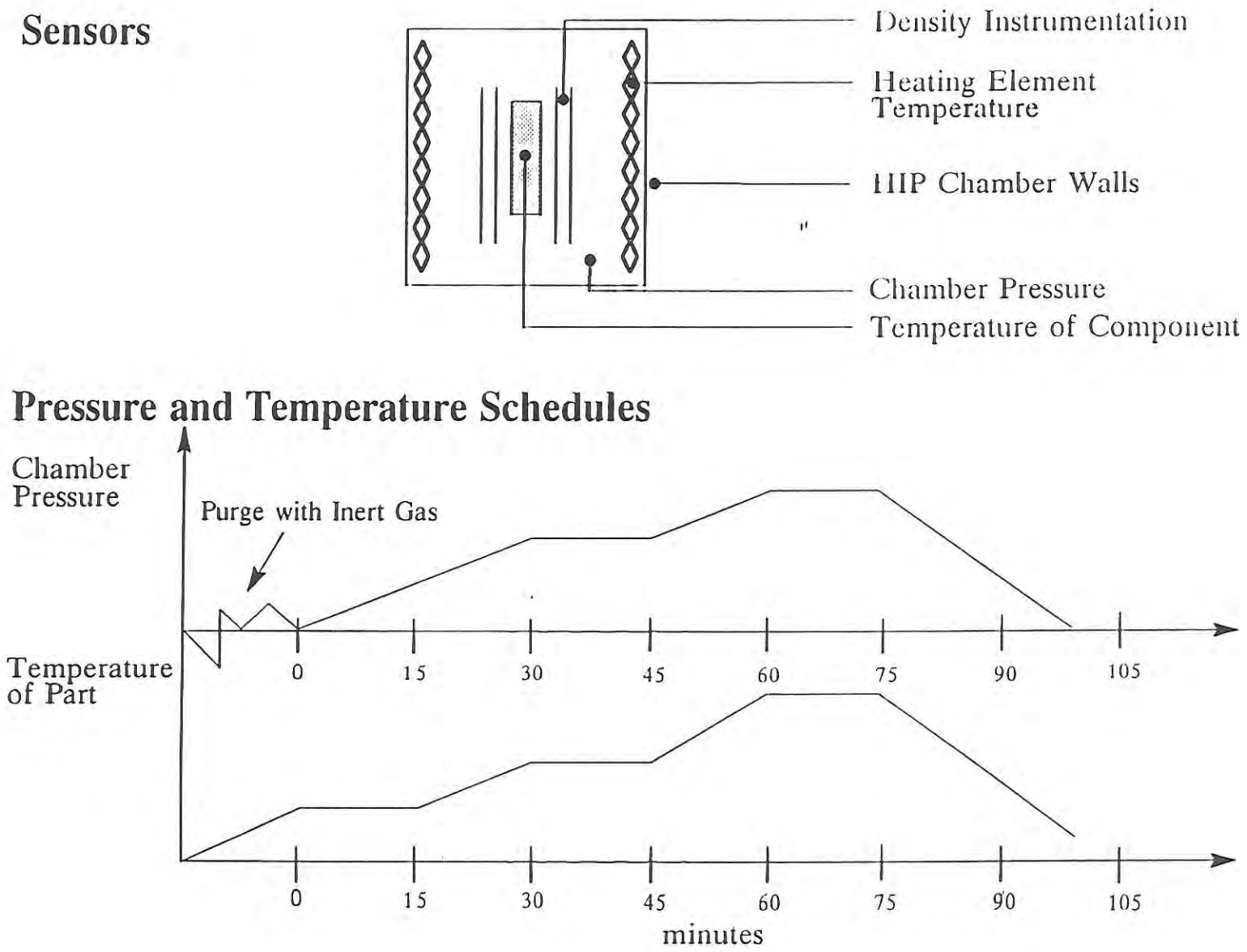


Figure 3. HIP Chamber and Process Schedule

The computer-based workstation is at the heart of the development of the intelligent controller. The workstation provides a rich information environment for the HIP expert to enhance the expert's control of the process. The new control heuristics may then be instantiated in a knowledge-based system. Figure 4 shows the computer network that will comprise the Phase I HIP workstation. Large CRT screens will display easily-used, understandable graphics of the sensor information coming from the HIP chamber, in conjunction with process simulation code that can be run from arbitrary initial states. Data logging and data base retrieval will be automated to free the user of administrative tasks and to enable him to focus on control of the HIP process. Color and three-dimensional representations of HIP process parameters, such as temperature, pressure, and density, will further increase the power of the expert. An ultrasonic *in-situ* sensor to measure grain size will be added to the functionality of the workstation as the technology of the sensor is developed. The workstation will form a man-machine interface that allows the human to act as predictor/corrector to achieve process control. Figure 5 shows a HIP control scenario.

The BDM Corporation will develop the workstation and conduct the knowledge engineering. It is expected that heuristics to avoid failure modes will come from experimentation and advice from the consortium. The goal of Phase I is to demonstrate the efficient production of titanium aluminide in cylindrical shapes with control over failure modes and material microstructure by man-in-the-loop process control. In Phase II, the intelligent controller will be designed to automate the control actions of the human.

Phase II of IPM/HIP will build on the expanding knowledge base in an environment that addresses the essential scaling issues of HIP at the production level — a microfactory. The dual purpose of a microfactory is to start producing urgently needed titanium aluminide intermetallics economically while providing a showcase for commercialization by American industry. The location of the microfactory will be chosen from one of the sites of the Phase I consortium members. Figure 6 depicts the completed Intelligent Control computer network.

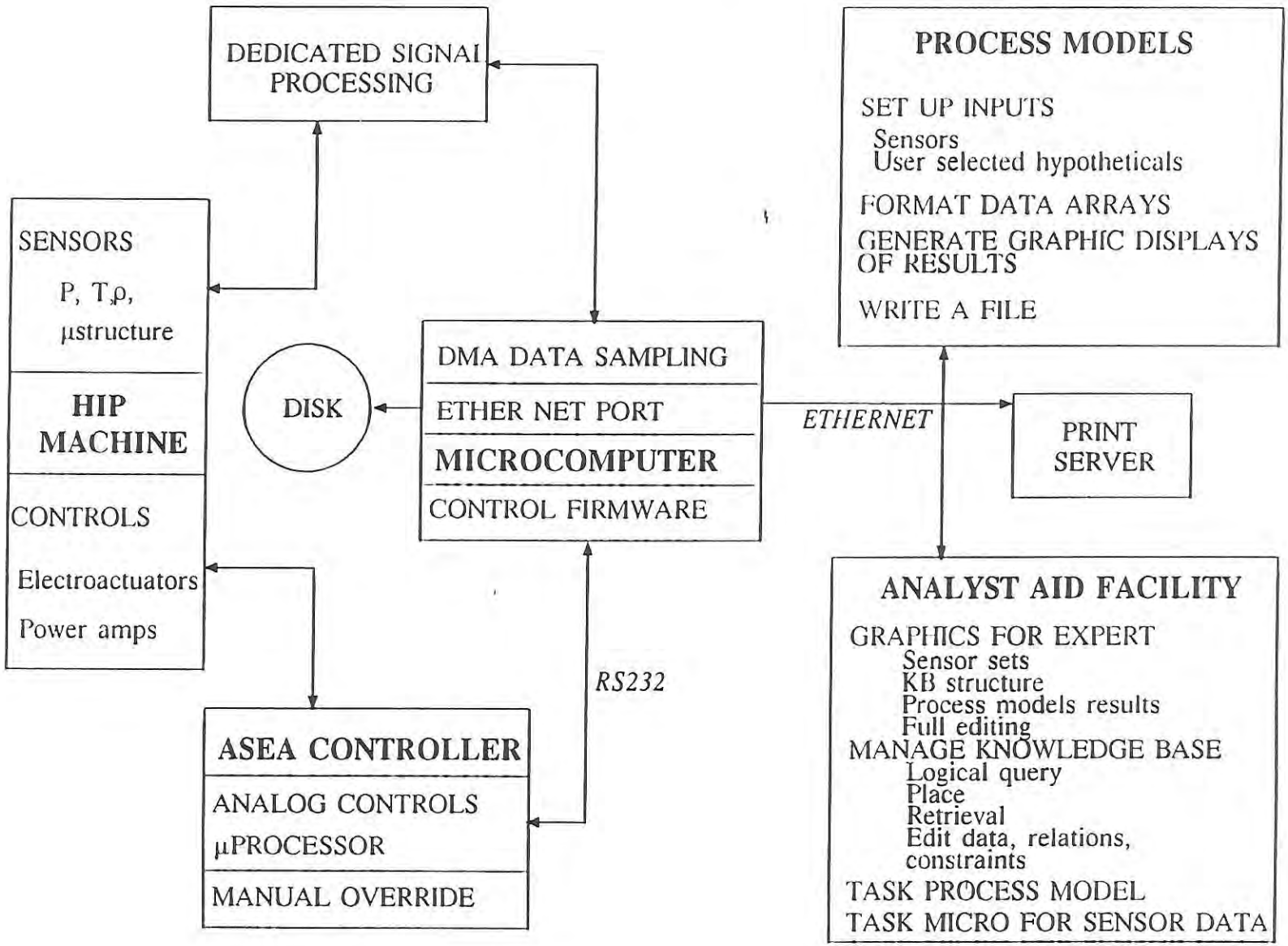
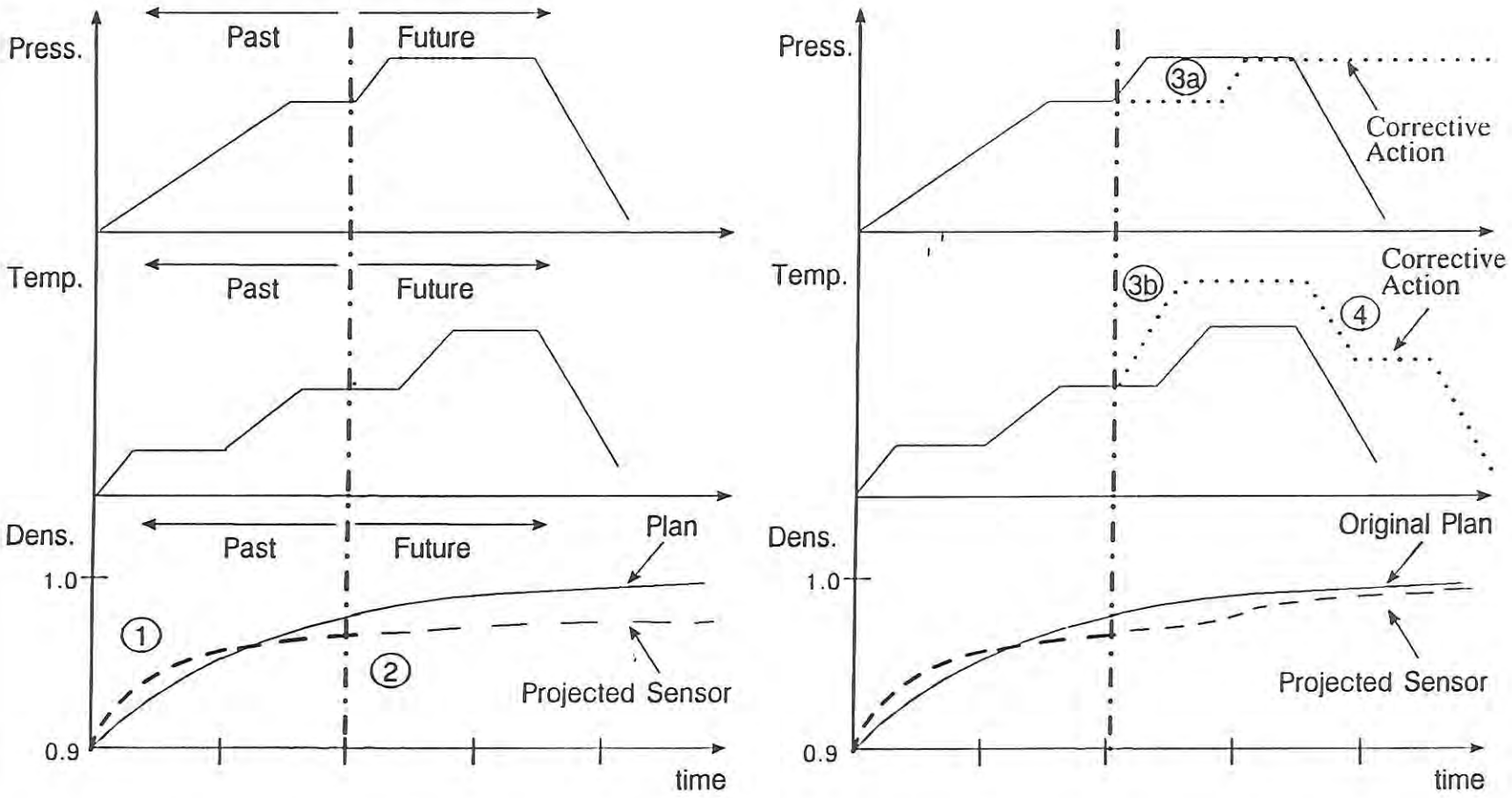


Figure 4. Computer Network for the Intelligent Control of the HIP Process



1. Sensing during the early stages of consolidation revealed a density front forming more rapidly than expected.
2. Analysis of the process state showed that, if the process were allowed to continue along the present schedule, the final density would most likely fall below the targeted level.

3a. Replanning by the intelligent controller calls for the immediate suspension of the scheduled pressure rise.

3b. The rate of heating is simultaneously to be increased to achieve a higher temperature level than originally planned in order to limit the consequenses of the densification front phenomena.

4. The intelligent controller further modifies its corrective actions by taking microstructure constraints into account. These constraints dictate that temperature must be reduced at the earliest opportunity consistent with elimination of the density front phenomena.

Figure 5. Example of Replanning HIP Control During Processing

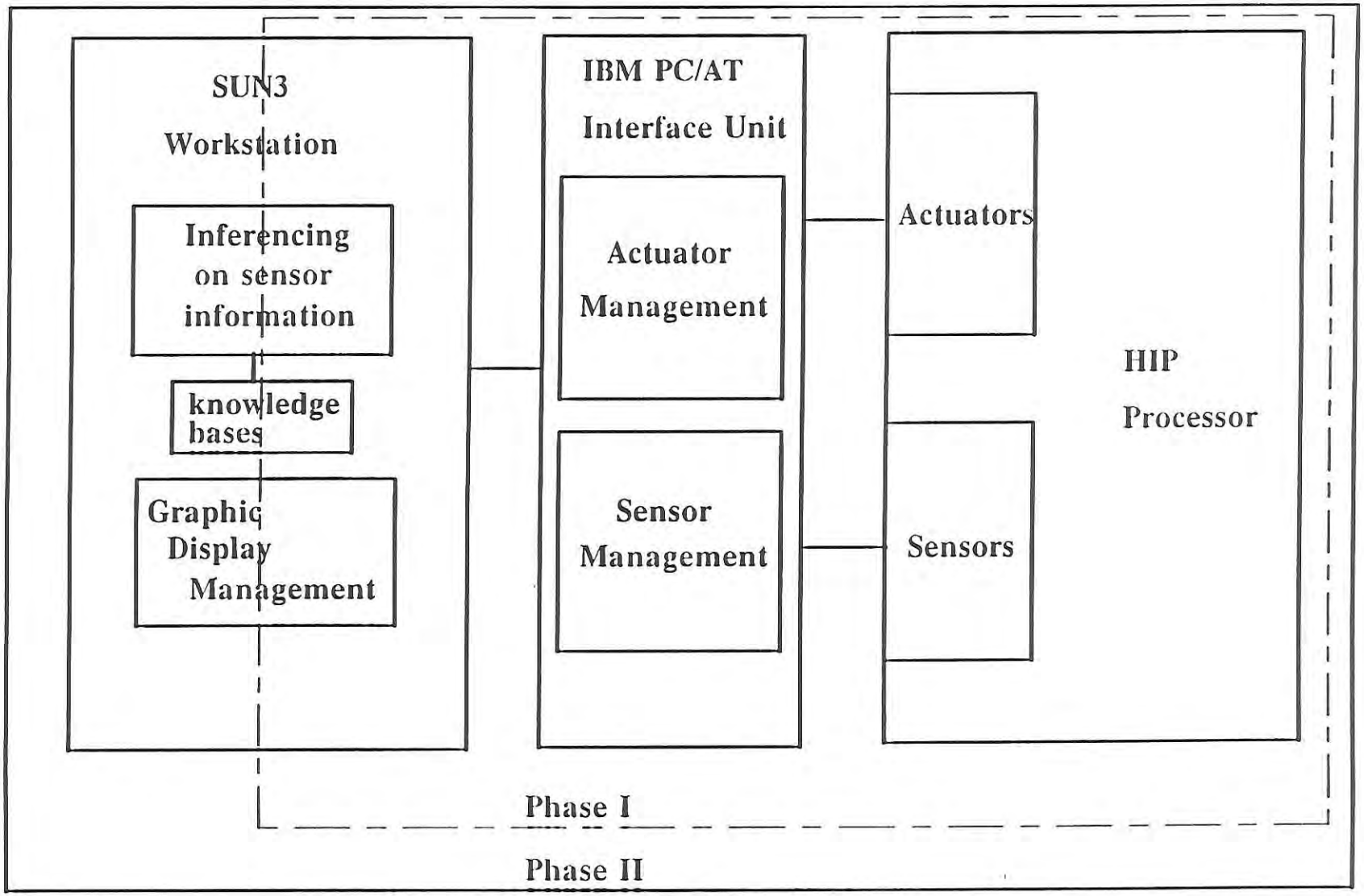


Figure 6. The Workstation Approach to Development of the Intelligent Controller

4.0 SUMMARY

The Intelligent Processing of Materials Program (IPM) is an interdisciplinary research thrust initiated by DARPA during 1986. It is aimed at capturing the expertise of the laboratory materials scientist and the process engineer to improve the quality, reliability, and yield of the new materials that are dictated by the latest DoD specifications. The IPM program will pursue the reduction in the historically long lead times between initial material development and eventual materials production and application. In doing so, the IPM program seeks to combine the use of expert systems, advanced *in-situ* sensors, and process models to control the quality of materials in a production environment.

It is envisioned that expert systems will have a significant impact on the IPM program and that the objectives of the IPM program cannot be met without experts systems. Knowledge acquisition is a critical activity in technical program due to different types of knowledge that are required in a single expert system, and due to the complexity of the domains selected for IPM application. The coupling of inputs from one or more knowledge sources, therefore, is clearly a formidable knowledge acquisition task, it is equally clear that the knowledge engineer is not adequately qualified to resolve the conflicts that will arise between domain experts. Much of the conflicts that will arise between domain experts. Much of the conflict resolution, therefore, must be accomplished by the experts themselves with the aid of an automated system that supports multiple representations of knowledge objects. This implies that some sort of intelligent system is required for the knowledge acquisition/knowledge engineering process.

ACKNOWLEDGMENTS

The authors wish to thank Dr. C. B. Friedlander and Dr. B. G. Kushner for their guidance and assistance in developing this paper and the systems described herein. Portions of the work described in this paper are supported by the Defense Advanced Research Projects Agency (DARPA) and the Army Research Office under contract number DAAL03-87-C-0019, and by DARPA and the Air Force Office of Scientific Research under contract F49620-86-C-0036.

BIBLIOGRAPHY

1. Firstenburg, A. et. al., "Research on Intelligent Processing of Carbon-Carbon Composites," Proposal to the Office of Naval Research, April 1987.

2. Helle, A. S., Easterling, K. E., and Ashby, M. F., "Hot Isostatic Pressing Diagrams: New Developments," *Acta Metallurgical*, Volume 33, 1985.

3. Kahn, A. H., and Mester, M. L., "An Eddy Current Sensor for the Measurement of Resisyivity and Temperature of Aluminum Rod During Extrusion", Presented at *Progress in Quantitative Nondestructive Evaluation*, June 1987.

4. Kushner, B. G., and Parrish, P. A., "The Intelligent Processing of Advanced Materials" *Intelligent Processing of Materials and Advanced Sensors*, AIME-TMS, 1987.

5. Kushner, B. G., Geesey, R. A., Parrish, P. A. and Wax, S. G., "A Knowledge Acquisition Tool for the Intelligent Processing of Advanced Materials", *Proceedings of the 1986 TMS/ASME Conference on Applications of Artificial Intelligence to Materials Processing*, TMS/ASME Press, 1987.

6. Loomis, W., et. al., "Analytical Processing for Improved Composites (APIC), Final Report," AFWAL-TR-81-4082, 1987.

7. Reeker L. H., T. A. Blaxton, and C. R. Westphal, "Applying Software Engineering to Knowledge Engineering (and Visa Versa)", to appear in the *Proceedings of the 27th Annual Technical Symposium*, June 1988.

8. Szekely J. and D. Apilian, eds., "Rapid Solidification by Plasma Deposition," Plasma Processing and Synthesis of Materials Symposium, 1984.

9. Wood, R. T., Intelligent Processing Initiative for Gallium Arsenide Crystal Growth, Second Quarterly Report, April 1987.

10. Yeheskel, O. and Y. Gefen, "Properties and Microstructure of Hot Isostatically Pressed Si_3N_4," *Materials Science and Engineering*, Volume 78, 1986.

Applying Hypertext Technology to Integrate Multiple Reasoning Models in a Knowledge Acquisition Tool

Karen L. McGraw, Juliana Lancaster, and Christopher Westphal

The BDM Corporation, Advanced Technology Group

1300 N. 17th St., Suite 950

Arlington, VA 22209

(703)247-0391

kmcgraw%gawain@bdm.com

Abstract

Expert systems that will assist in the solution of C^3 problems will require the use of multiple domain experts. The requirement to tap multiple experts results from the fact that in complex domains, any single expert may be very knowledgeable about only a small subset of the domain. Even when a single expert embodies required domain knowledge, the use of one expert for knowledge acquisition in highly dynamic domains may not produce knowledge base content that is diverse enough to accommodate anticipated domain problems. Using more than one expert enables developers to make use of different problem solving approaches, considerations, and applications of knowledge. Multiple viewpoints can be incorporated in a "multiexpert" expert system such that each line of reasoning can contribute unique information that others can access and use. The BDM Corporation has been investigating the use of hypertext technology to integrate and represent multiple (possibly conflicting) experts' views within BDM-KAT, a knowledge acquisition tool. The application of hypertext to the multiple expert problem provides a powerful vantage point from which to acquire, represent, and access expert knowledge, while preserving dissenting opinions. These techniques may also be integrated into other applications in which the preservation of multiple opinions is essential, such as intelligent tutoring systems, intelligent computer managed instruction and decision aiding systems in the C^3 arena.

Applying Hypertext Technology to Integrate Multiple Reasoning Models in a Knowledge Acquisition Tool[1]

Karen L. McGraw, Juliana Lancaster, and Christopher Westphal
The BDM Corporation, Advanced Technology Group
1300 N. 17th St., Suite 950
Arlington, VA 22209

Introduction

Expert systems that will assist in the solution of C^3 problems will require the use of multiple domain experts. As a domain, the C^3 arena is particularly susceptible to problems resulting from multiple expert input. Specifically, typical C^3 problems will require that knowledge-based decision aiding systems be able to handle and/or integrate knowledge from experts who may represent diverse fields or subdomains. Furthermore, addressing this issue requires that the system be able to either resolve conflicting opinions and/or document their occurrence. The combination of emerging technologies, such as knowledge acquisition tooling and hypertext or hypermedia provides an approach to managing multiple expert input and multiple reasoning models within a domain. This paper describes one example of this approach and the synergy that can result from depicting and accessing multiple reasoning models. First, we describe BDM-KAT, a knowledge acquisition tool developed to capture process knowledge from the materials processing domain. We illustrate how this tool helps a knowledge engineer elicit a mental model for the domain by describing the different tools and modes available. Next, we detail the importance of basing a knowledge based system on valid mental models and discuss the need for multiple reasoning models. Finally, we describe the use of hypertext/hypermedia technology to enable the depiction and use of alternate or additional models and/or supporting text.

Knowledge Acquisition Tooling

Knowledge acquisition has been referred to as "the bottleneck in development," "the most problematic aspect," and the "most essential component of expert system development." Several problem areas within current knowledge acquisition methodologies contribute to this assessment (McGraw & Harbison-Briggs, 1989):

- Inability of domain experts to verbalize decision making processes
- Difficulties relating to the conceptualization of an expert's mental model for a domain
- Inaccuracies inherent in the communication process, on which traditional knowledge acquisition is based
- Difficulties involved in translating accurately the knowledge acquired from a session into knowledge base code.

[1]Work described in this paper is supported by the US Defense Advanced Research Projects Agency, through the Air Force Office of Scientific Research under Contract number F49620-86-C-0036 and the Army Research Office uncer Contract number DAAL03-87-C-0019. The views and conclusions contained in this paper are those of the authors and are not to be interpreted as representing the official policies, either expressed or implied of the Defense Advanced Research Projects Agency or the US Government.

To address these problems, the BDM Corporation's Advanced Technology group is developing the Knowledge Acquisition Tool (BDM-KAT). Although BDM-KAT currently is aimed at the acquisition of process knowledge, it is designed to overcome some general problems associated with knowledge engineering in a broad variety of knowledge-based applications. Using tools within BDM-KAT a knowledge engineer is able to elicit knowledge from experts in a structured manner such that the knowledge acquisition bottleneck is loosened. The end product of a series of sessions is a knowledge base for an expert system or decision-aiding system.

Working with BDM-KAT the knowledge engineer and domain expert interact with several specialized tools to build a process model for a domain. Tools currently in design and implementation for this purpose include: (1) Rule Builder, (2) Causation Charter, (3) Categorizer, and (4) Time Mapper. These tools stimulate the elicitation of process knowledge in a manner that reduces typical knowledge acquisition risks. The process model that results from iterative use of these tools is developed and refined in three modes: Clarification, Prediction, and Diagnosis. In the Clarification mode the knowledge engineer elicits declarative knowledge about the domain (Figure 1). In this mode, primary objects in the expert's domain are detailed and their relationships made explicit. The expert is provided with facilities for creating a graphical representation of the process. The objects that are built into the model are represented in frames in a hierarchically-organized knowledge base (Reeker, Blaxton, & Westphal, 1988).

After the initial model for the domain has been created, the Prediction mode can be evoked to provide a process flow view of the model. Goals for this mode include setting parameters for each stage in the process model, exposing deficiencies in the model, and allowing additional information to be inserted (Blaxton & Westphal, 1988). In this mode the domain expert and knowledge engineer identify sensors that are used in the process and develop active images for those sensors (e.g., thermometer, clock, flow rate monitor, pressure

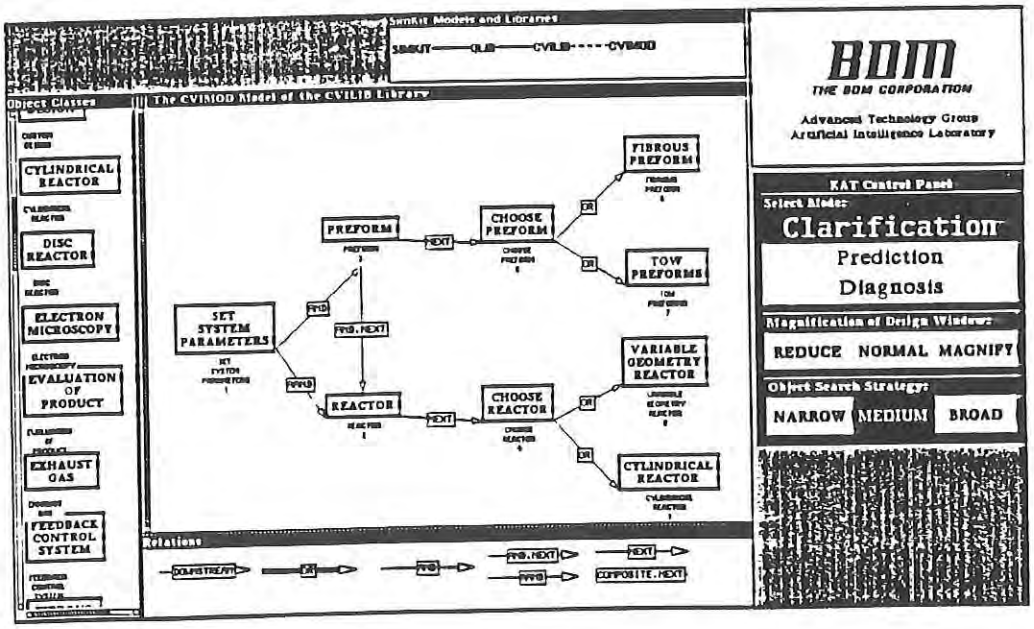


Figure 1. BDM-KAT's Clarification Mode

gauge). Once developed, value ranges for each sensor are set (i.e., high, average, and low readings are defined). Additionally, the expert is asked to anticipate potential faults at each given point in the process. Problems in obtaining responses to these requests are minimized by having the expert interact directly with the active image that represents a crucial sensor. As it is likely that this identification of parameters, sensors, values, and potential problems will prompt the expert to redesign aspects of the process, iterations between Clarification mode and Prediction mode are expected.

The result of iterations between these modes is a process model that represents a fairly detailed knowledge structure of the domain's concepts, conceptual hierarchies, facts, and heuristics for using those facts. At this point the knowledge engineer and domain expert can use the Diagnosis mode to step-through a presentation of the process that has been detailed. Much like an event-stepped simulation, this mode enables the domain expert to examine the model he or she has developed in a more active way. Later, key values and parameters of the model are challenged through selected perturbations to the model. This activity (1) exposes nuances and inaccuracies that may have remained undetected in the previous modes and (2) enables controlled iteration through the previous modes as the model is corrected and expanded (Westphal, 1988).

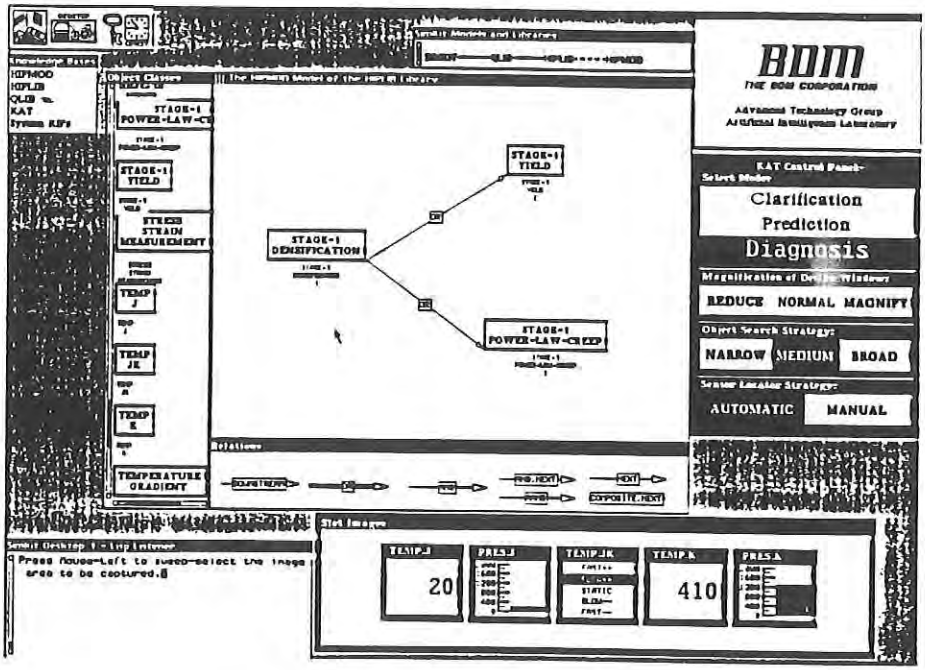


Figure 2. BDM-KAT's Diagnosis Mode

Mental Modeling in a Domain

The process model developed through the interaction between an expert and BDM-KAT will not necessarily reflect a "textbook" version of the domain. Rather, it will represent the expert's **mental model** of the problem and domain under consideration. While it may seem that an expert system would be better served by a knowledge base that is compiled from known scientific facts about the domain, there is much about the ways in which we reason with and from knowledge that this approach would not capture. For example, the concept

of a script has developed from the awareness that we approach the world with expectations and beliefs that we use to guide our behavior, our interpretation of what we see and hear, and our developing plans for reaching a goal or solving a problem (Mandler, 1979; Schank, 1977). The plans we generate based on our expectations are not static or fixed, but rather are easily adaptable for changing situations (Gick & Holyoak, 1983; Hammond, 1986; Kolodner, 1985; Kolodner & Simpson, 1984). This implies that the models on which the plans are based are not static, but can be reasoned with and manipulated. To achieve this capability in knowledge based systems, we must develop knowledge bases that are not just static representations of the domain content, but that represent usable models of the domain (i.e., we must capture the expert's own mental model). The knowledge bases elicited by BDM-KAT are created, structured, and modified by the expert using various tools to translate his or her own mental models.

The Need for Multiple Reasoning Models

Developing a knowledge acquisition tool to capture a single model for reasoning or processing is useful as it allows developers to frame the process, provide a common reference point, and illustrate the various tasks and subtasks involved. As a model in BDM-KAT is clarified and enhanced, for example, the characteristics of its components are simultaneously translated into code that becomes the initial knowledge base. However, in complex domains such as C^3, developers might not find it acceptable to build the knowledge base upon a single mental model. These domains often require multiple reasoning models for the following reasons, each of which will be discussed briefly:

- A large domain may be composed of several subdomains, each having its own required expertise;
- Multiple domain experts may approach problem solving within the domain from slightly different perspectives, even though the facts they refer to are similar; or
- Experts in a given domain may have non-compatible models of the structure, content, and/or implications of the domain.

Separable Subdomains. The requirement to tap multiple experts is due partially to the fact that, in complex domains, any single expert may be very knowledgeable about only a subset of the domain (Mittal & Dym, 1985; McGraw & Seale, 1988). Knowledge within a complex domain may exist in various separate but interrelated "subdomains." Problem solvers in the domain access and use knowledge from each subdomain to accomplish specific tasks; yet each piece of information may provide only part of the solution to the problem. In applications for which multiple subdomains are evident or in which experts disagree as to the structure of a process model it would be useful to be able to depict and/or invoke portions of an alternate model. For example, for the Mentor system developers were able to identify several experts in the servicing of refrigeration equipment, each of whom had a particular specialty (Cochran and Hutchins, 1987). Similarly, developers of the DARN project (Mittal, Bobrow, and deKleer, 1984), a system to assist in computer maintenance, identified a variety of computer maintenance personnel early in the development process. They quickly realized that although the selected maintenance individuals all had a *general* understanding of the problem area, each had his or her specialty area within the general domain. In a system as complex as the Pilot's Associate, subdomains that contribute to a course of action may be even more dissimilar, requiring knowledge from subdomains such as system status, situation assessment, mission planning, tactics, and pilot vehicle interface.

Multiple Perspectives. Even if we were able to identify a single individual who embodies most of the required domain knowledge in a particular area, the use of one expert for knowledge acquisition in highly dynamic domains may not produce knowledge base content that is diverse enough to accommodate anticipated domain problems. Using more than one expert enables developers to make use of different problems solving approaches, considerations, and applications of knowledge. Previous research by LeClair (1985) has determined that multiple viewpoints can be incorporated in a "multiexpert" expert system, such that each line of reasoning can contribute unique information that others can access and use. Thus, system users can work from the combined strength of different problem solving models or solution sets.

Non-Compatible Models. Since knowledge acquisition is still an intensely interpersonal communication problem, tools built to assist in this task must be able to graphically present the developing mental model to the domain expert. This presentation is a form of feedback, in which the expert uses the tool to verify domain knowledge elicited earlier. In the case of a single expert, this verification usually leads to refinement, but may not result in dramatic changes in the structure of the model. However, in the case of multiple experts, it may be difficult for one expert to view and agree with the structure of a model created by another domain expert, simply due to the fact that different users may have different mental models of a system's operating processes (Lehner, Rook, & Adelman, 1984) or may be using different labels to depict it. The degree to which contributing experts share the same lowest level data items, objects, and properties with the process model determines the cognitive consistency of the model with the human view of processing or reasoning in the domain. Being able to represent differences for later exploration and consideration helps developers increase the validity of the model on which the system will be based and thus, the validity of the knowledge base itself.

Hypertext for Multiple Models

Hypertext/hypermedia is a technology for "building systems for information representation and management around a network of multi-media nodes connected together by typed links" (Halasz, 1988, p. 836). Hypertext technology provides mechanisms that can be used to integrate and represent multiple (possibly conflicting) experts' views. This technology enables non-monotonic "idea processing" by providing

- Windows on the screen that are associated with objects in a knowledge database of objects (e.g., domain concepts and categories)
- Links or relationships between these objects
- Chunking of information into small units or nodes
- User determined navigation through the database.

Current hypertext systems support both graphics and fully-formatted text nodes. Users can browse contents within the "database" by navigating through the system. This is generally accomplished by selecting the target about which more information is desired. The application of this technology to the multiple expert problem provides a powerful vantage point from which to acquire, represent, and access expert knowledge, while preserving dissenting opinions. As an expert views a process model created in BDM-KAT he or she can add to or provide alternate models that may result from past experience, a different perspective, or the particular context of a problem. The expert can then call up one or more windows and provide (1) text explaining exceptions to a model, (2) considerations for the model's use, and/or (3) a graph of an alternate model (or portion thereof). The problem

solving environment that results is one that enables experts to compare, contrast, and follow different viewpoints (e.g., models, processes).

The representation of, and access to information supplied to BDM-KAT by multiple experts is accomplished through an interactive, inspectable "hyper-expert" facility. A process frame of the domain model's base structure may contain an active, graphic icon (i.e., a crosshatch) indicating that additional information about this frame has been provided by another expert. Upon selecting this icon a user receives a menu that displays titles to all nodes relating (e.g., linked) to that icon. Each entry is an active element that when selected, displays a window into the expert's comments (e.g., justification or explanation for differing opinion) and/or a "sketchpad" containing a model of that expert's view. The explicit relationships among the representations of the experts' knowledge allow the interpolation of new reasoning strategies and browsing.

Conclusions

The use of knowledge acquisition tooling and hypertext technology as a partial solution to the problem of multiple experts can be a significant step toward the construction of systems that enable reasoning from multiple models. The developers of these systems recognize that mental models are inextricably tied to problem solving context and individual experts. Providing facilities to allow the elicitation of mental models (and variations of portions of the model) enable users to browse the system and select the most appropriate reasoning model for the problem at hand. Not only can the elicitation of models and their variations enable browsing to *select* that which is most appropriate, but browsing also may result in extensions of an expert's understanding of the domain as he or she examines other ways of looking at a process.

References

Blaxton, T.A. and C.R. Westphal (1988) Explicit queries with simulation techniques during knowledge acquisition. *Procedings of the 1988 Eastern Simulation Conference on AI and Simulation*, Orlando, FL., 17-22.

Cochran, E. and B. Hutchins. (1987) Testing, verifying, and releasing an expert system: The case history of Mentor, *Proceedings from the Third Conference on AI Applications*, Kissimmee Florida, February 23-27, 1987.

Gick, M.L. & K. Holyoak. (1983) Schema induction and analogical transfer. *Cognitive Psychology, 15*, 1-38.

Halasz, F. G. (1988) Reflections on NoteCards: Seven issues for the next generation of hypermedia systems, *Communications of the ACM, 31(7)* 836-852.

Hammond, K. (1986) Chef: a model of case-based planning. *Proceedings of the National Conference on Artificial Intelligence*, Philadelphia, PA.

Kolodner, J. L, (1985). Expertise in problem solving. Technical Report #GIT-ICS-85/23. School of Information and Computer Science, Georgia Institute of Technology, Atlanta, GA 30332.

Kolodner, J. L. & R. Simpson (1984) Experience and problem solving: A framework. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, Boulder, CO.

LeClair, S.R. (1985). A multiexpert knowledge system architecture for manufacturing decision analysis, published dissertation, Arizona State University.

Lehner, P., F. Rook, and L. Adelman. (1984) Mental models and cooperative problem solving with expert systems. PAR Technology Corporation Report AD-A147-843.

Mandler, J.M. (1979). Categorical and schematic organization in memory. In C.R. Puff (ed), *Memory organization and structure*. New York: Academic Press.

McGraw, K. and K. Harbison-Briggs (1989). *Knowledge engineering: Principles and guidelines*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

McGraw, K. and M. Seale. (1988). Knowledge elicitation with multiple experts: Considerations and techniques, *Artificial Intelligence Review, 2(1),* 29-42.

Mittal, S. and C. Dym. (1985) Knowledge acquisition from multiple experts, *The AI Magazine,* 32-36.

Mittal, S., D.G. Bobrow, and J. deKleer. DARN: A community memory for diagnosis and repair tasks. Unpublished report, Xerox PARC, Palo Alto, CA.

Reeker, L.H., T.A. Blaxton, & C.R. Westphal (1988) Applying software engineering to knowledge engineering (and vice-versa). *Proceedings of the 27th Annual Technical Symposium of the Washington, D.C. ACM chapter,* Gaithersburg, MD, 107-114.

Schank, R. (1977) *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Erlbaum.

Westphal, C.R. (1988) Using simulation-like methodologies for knowledge base validation. *Proceedings of the Third International Symposium on Knowledge Engineering,* Madrid, Spain.

APPLYING SOFTWARE ENGINEERING TO KNOWLEDGE ENGINEERING (AND VICE-VERSA)

L. H. Reeker, T. A. Blaxton, and C. R. Westphal

The BDM Corporation, Advanced Technology Group
7915 Jones Branch Drive, McLean, VA 22102

Abstract

An underlying thesis of this paper is that software engineering at the program specification level is closely related to knowledge engineering for expert systems describing processes, and that practitioners of each can benefit substantially through comparison of techniques and objectives. It is pointed out that a computer program is a process and that the activities described by computer programs are processes. Sometimes they are problem solving processes, but increasingly often, they are difficult to conceptualize in the problem solving paradigm. Within this context, a knowledge engineering tool for processes is described and discussed from two viewpoints. First, the tool itself is the embodiment of a number of principles and methodologies of software engineering. Second, it is argued that such a tool can be useful in software engineering, particularly in the cases where the software is not easily conceptualized as problem solving.

1. Introduction: Process Knowledge and Programming

The enterprise of software engineering is dedicated to creating programming methodologies and environments that facilitate the development of maintainable and verifiable software. A critical determinant in the design of tools to aid the software engineer is the manner in which the software engineering activity itself is conceptualized. Programming has traditionally been regarded as a problem solving activity. Thus, software engineering emphasis has often been on developing software aimed at solving particular classes of problems, even where the applications are not readily characterized as problems, in the sense of having initial states, solution methods, and solution states. In these latter cases, we argue that it is useful to think of a computer program as the specification of a process. The idea that a program specifies a

Work described in this paper was supported by the U. S. Defense Advanced Research Projects Agency, through the Air Force Office of Scientific Research under contract number F49620-86-C-0036 and the Army Research Office under contract number DAAL03-87-C-0019. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U. S. Government.

process will not seem radical and may be seen as implicit in certain types of programming, particularly in simulations; but it represents a shift of emphasis from the tradition of representing a program as a problem solving method. Processes may be problem solving activities, of course, but they may be many other things, including real-time control and operating systems. For instance, many of the examples cited as "components of complex systems" by Winograd (1979) are most easily conceptualized as processes (or as behavioral descriptions of systems), than as problem solving activities.

Once the model of programs as processes is accepted, the software engineering researcher's task becomes one of studying and providing facilities for the specification of processes in a disciplined and natural manner. In this paper we describe a tool which provides such facilities. This tool was designed not as a software engineering environment but as a system for acquiring knowledge about processes. The Knowledge Acquisition Tool (KAT*) is being built using some important principles of software engineering, but it is clear that the techniques embodied in KAT for knowledge engineering can be fed back into the design of programming environments for software engineering.

The mutually beneficial relationship between software engineering and knowledge engineering will be illustrated in two steps. First, we describe the operation of KAT, highlighting principles and techniques from software engineering that have been incorporated into its design. Second, we reverse our perspective and describe how techniques for acquiring process knowledge in KAT can be used to design better software engineering tools.

*The acronym KAT has proven irresistible for a knowledge acquisition tool. We have become aware of others in the literature, so the user may wish to read this not as *the* KAT, but as *a* KAT. Specifically, we have begun calling it BDM-KAT.

2. KAT – A Knowledge Acquisition Tool for Processes

It has become apparent that one of the major obstacles in applying artificial intelligence to specialized domains is the difficulty of gaining from human experts the knowledge necessary to create large and detailed knowledge bases. One approach to widening this "knowledge acquisition bottleneck" is to design computerized tools to elicit information from domain experts. KAT is being developed for this purpose.* The initial application area for which KAT is intended is materials processing. KAT is expected to expedite the transition of techniques for the synthesis of new materials from the laboratory to production environments. Although KAT is specifically aimed at the acquisition of process knowledge, it is being designed to overcome some general problems associated with knowledge engineering in a broad variety of knowledge-based AI systems.

To understand the need for building a computerized tool such as KAT, it is only necessary to consider the difficulties associated with the construction of knowledge bases. Figure 1 shows the typical life cycle of a knowledge-based system as a variation on the more general software life cycle (Jackson, 1975). Considerable effort is expended in knowledge acquisition and prototype testing, largely because information gleaned from human experts is often either incomplete or incorrectly coded in early prototype versions of the system. This is true in part because the typical approach to building knowledge bases entails prolonged and intensive one-to-one interactions between a programmer and a domain expert. Thus there is an extra translation step between the expert and the implementation which may result in the misrepresentation of knowledge and the introduction of bias. Traditional methods of knowledge engineering are unstandardized and provide no formal process specification for knowledge base verification.

*KAT is currently in prototype development stage. By prototype development, we mean that specifications are complete and an initial demonstration system designed for a particular domain (chemical vapor infiltration) is implemented, but that a large-scale deliverable prototype to be developed under the contract will not be available for several months.

We seek to rectify these problems in KAT by providing a structural format which remains consistent throughout each application. The reader should note that although KAT will automate the knowledge acquisition process to a large extent, it is not intended to replace the knowledge engineer. Rather, it is designed as a tool to aid the engineer in developing a knowledge base.

2.1. A Closer View of KAT

KAT is intended to address the dotted knowledge engineering portions of the diagram in Figure 1, providing a medium for experts to express relevant information about a processing domain. This knowledge is captured and encoded in appropriate internal representations (i.e. objects, frames, rules) by cycling through different modes of query designed to elicit knowledge from the expert in an organized and efficient manner.

Using KAT, the knowledge engineer and domain expert iterate through the three querying modes indicated in Figure 2. In each mode, the expert is prompted to answer different types of questions about a particular materials processing domain. In the **Clarification Mode** (Figure 3), the expert is provided with facilities for creating a graphical AND/OR representation of the process. Model objects, or data items, included as components in the graph will be represented in frames in a hierarchically organized knowledge base. Clarification is thus designed to elicit primarily declarative knowledge. The user will place objects in the graph and assert logical relations (AND, OR, and NEXT) between them.

The **Prediction Mode** (Figure 4) provides a process flow view of the model, exposing "holes" that were missed in the Clarification Mode, allowing additional information to be inserted as necessary. Time and sequencing relationships that may only be implicit in the AND/OR representations are shown explicitly in this mode, though the logical design history explicit in the AND/OR graph is suppressed in the creation of composite objects. These composite objects are displayed with simple "next" relations in the flowchart format, emphasizing

the temporal flow of the process.

It is emphasized that the flowchart is *not* used to compose the process specification, but rather to check the specification previously composed using a structured methodology. It provides another perspective on the specification, providing a more refined framework for posing certain questions that reference process sequences. In the Prediction Mode, the expert is asked to anticipate potential faults at each given point in the process, and to provide suggestions for redesign that might lead to avoidance of identified problems. Information gleaned from some of these queries is recorded into the knowledge base in rule form, though some knowledge can be maintained as "comment" text. Once these queries are completed, the knowledge engineer may return to the Clarification Mode and recycle through the questioning, adding to the knowledge base created thus far.

Once iteration between Clarification and Prediction Modes has occurred a few times, a fairly detailed knowledge structure will be in place. At this point, the **Diagnosis Mode** (Figure 5) is invoked. In this mode, the KAT facilities provide a step-through presentation of the materials process as specified up to this point, much like an event-stepped simulation, with the depth of the simulation limited to the knowledge imparted by the expert*.

2.2. Software Engineering Techniques in KAT

Although KAT may appear to be an expert system in the acquisition of materials processing knowledge, its current structure is closer to that of a software design environment. The KAT prototype is being built on a Symbolics 3670 in the KEE and SimKit environments. KEE is used because it provides many facilities to aid AI programmers in the generation of frame-like objects and rules. In addition, it has graphics and specialized mouse functions which are helpful in constructing user interfaces. The SimKit system, which resides on top of

*For an overview of simulation in the knowledge acquisition process, see (Westphal, 1988).

KEE, provides a host of facilities for writing event-driven simulations, including a clock, event calendar, data collectors, and various mathematical tools for data analysis (IntelliCorp, 1986). The event-driven nature of the SimKit simulation environment makes it ideal for the materials processing domain.

KAT is not currently intended to produce an expert system; rather, it is to be used as a means of producing a knowledge base by using both top-down and bottom-up process knowledge elicitation methodologies. The top-down portion, embodied in the Clarification Mode, is based on a technique akin to the Higher Order Software approach (Hamilton and Zeldin, 1976; Harel, 1980) used in complex software development for NASA, extended slightly to allow the specification of parallel processes (Reeker, 1986). The bottom-up portion, found in the Prediction Mode, and in dynamic form, in the Diagnosis Mode, features flowcharting that can be rearranged dynamically and transformed back to the top-down AND/OR representation. The facilities provided in the Prediction Mode are not unlike those of TASC's Adagraph product, based on the Pamela design methodology (Cherry, 1986). By iterating through the three query modes, the expert will be exposed to a variety of knowledge acquisition techniques (Figure 2).

The Clarification Mode (Figure 3) may be thought of as a framework that allows the expert to inform the system about the component subprocesses of each process, using a top-down or process decomposition viewpoint. As mentioned above, the result is an AND/OR graph. The OR relations, where deterministically specified, will contain branches labeled by conditions that determine the appropriate path to select. If the conditions are not known, they may be indicated by question marks to denote alternatives that have not yet been resolved or will be decided by later indications of success or failure (nondeterminism), or by the initials of one or more experts who may have specified different alternatives.

Once a certain amount of information is encoded in AND/OR format, the user moves into the Prediction Mode (Figure 4). This transition entails a conversion from the knowledge structure created during the Clarification Mode to a flowchart display. In order to perform this transition, AND and OR tree structures are collapsed down into single composite objects. The composites incorporating AND relations represent sequentially cohesive units, and those comprised of OR relations are logically cohesive (Parnas, 1972; Page-Jones, 1980). The composite objects may be expanded into their original parts as desired.

For each subprocess point in a Prediction Mode flowchart under the current system, the expert is asked to indicate sensors that are relevant to the process, and to specify their normal operating ranges. In addition, the expert is asked questions concerning the interpretation of various sensor readings at each subprocess point, including extreme readings. The responses to such queries take the form of inference rules. For example, the expert might state: "If the temperature of the reactive chamber is greater than 1200 degrees, the infiltration of the preform will be uneven". KAT provides mechanisms to obtain these responses through an interface that attempts to minimize for the user the problems associated with the syntactic and semantic issues of rule-based structures (Blaxton and Westphal, 1988). It is likely that this identification of potential problems will prompt the expert to redesign the process in some cases.

By the time the expert moves into diagnosis, the Clarification and Prediction Modes will have been iterated many times. The Diagnosis Mode (Figure 5) provides a systematic check of the knowledge structure to allow the domain expert to identify problems with that structure by observing errors in its performance during a limited process simulation (Suwa *et al*, 1982). The detail might initially be quite sketchy, but as more knowledge is added, it becomes increasingly explicit. Where it is not adequately explicit, the Diagnosis Mode may expose a need for more detail. This process is analogous to something between a static code check and

an interpretive execution. The control statements implicit in the clarification and prediction inputs of the expert are used for sequencing, but conditions that affect sequencing can be chosen interactively and a record kept, so that various alternatives are tried.

As may be seen in Figure 5, specialized graphics are used for displaying sensor readouts. In addition, the expert can observe directly the consequences of process rules defined during the Prediction Mode. If at any time the expert perceives the process to be going awry, the simulation can be halted and the system redesigned as necessary by returning to the other modes. A redesign could include addition, deletion, or rearrangement of subprocesses, rules, or objects. We have examined the requirements making the mode more dynamic in the materials processing case, and it is considerable, including qualitative physical knowledge (De Kleer, 1984; Kuipers, 1984). Though we have plans to develop this aspect of KAT in the future, the Diagnosis Mode currently consists only of the event stepped "tour" of the process specified, allowing the expert to verify that the events are taking place as specified and that conditions specified for the events are indeed true.

The most important point to note about KAT's three query modes is that they are specially designed to elicit different types of information from the expert. Recent research on human memory has shown that the information people retrieve upon a given query depends largely on the way in which that query is posed (e.g., Roediger and Blaxton, 1987; Schacter and Graf, 1986). In particular, it has been shown that explicit queries (such as those in the Clarification and Prediction Modes) elicit different types of information from that obtained by implicit queries (Diagnosis), as indicated in Figure 2. In addition, the change in format of model presentation across modes constitutes a variation on retrieval cues that should help to elicit otherwise unrecalled information, leading to a more complete specification. Weinberg (1971), among others, pointed out the utility of considering multiple views of programs. This

observation is valid both for code and for process specifications. Other systems such as EPOS (Biewald *et al*, 1979) use similar techniques of incorporating multiple views when transitioning between design levels. The iteration of clarification and prediction provides opportunities for the elicitation of different types of knowledge and their organized integration into a single knowledge structure.

3. Applying Knowledge Engineering Techniques to Software Engineering

We now consider the application of techniques used in the acquisition of process knowledge to the enterprise of software engineering. It is important to remember that we are discussing the whole software engineering process, from the formulation of a specification to the delivery of finished code. This includes debugging and documentation.

As was stated at the outset, it is beneficial to view computer programs as describing processes, rather than as solving problems. The tradition of problem-oriented software engineering practices has its roots in the development of early programming languages. For example, it was the methodology of problem solution that was captured in the early higher-level languages. ALGOL tells the story in its name: ''Algorithmic Language''. The emphasis has been on supplying inputs to an algorithm and obtaining outputs (solutions to the problem). Unless the term is stretched a good deal, however, problem solving is not the most common use of computers today. We are all familiar with operating systems, real-time control systems and others that do not fit well into the ''problem solving'' mold. All of these specify processes.

If one thinks of programming as process specification rather than as problem solving, the techniques available from work in the acquisition of process knowledge are available for use in creating better programming environments. The basic entities by which processes are described in a system for knowledge engineering about processes, such as KAT, include the

lowest level subprocesses and the structures that are referenced in the description of the process at any level. In using KAT to develop software specifications for programming a process, natural language descriptions of subsystems known to be programmable would constitute these lowest-level items. If KAT were being used to produce code directly, they would be statements in a programming language or programs already written and being reused. There should be present in the software engineering environment a variety of statements of a functional, imperative, and declarative nature that can perform the desired operations. It should be possible to create new ones within the environment by composition of processes and by definition of basic processes on data structures, thus encapsulating data in a desirable manner.

Additional basic entities in software engineering are the data structures used in programming. In today's software engineering methodology, data structures not built into the language are described by data abstraction techniques. In fact, this is exactly how their counterparts in knowledge engineering are described. In a navy knowledge-based system in which ships are basic objects, for instance, the system only knows what a ship is in terms of the operations that can take place on ships and the predicates that can be used to make queries about ships. A ship is a different data object in a system for naval personnel records from what it is in a system for naval battle management.

Additional treatment of knowledge representation in knowledge engineering as the analogue of data structures in software engineering can be found in (Cooke and McDonald, 1986). DuBois *et al* (1986) have developed an approach that emphasizes knowledge representation of the sort used in knowledge-based systems within a requirements engineering framework similar to that of (Greenspan and Mylopoulos, 1982). A suggestion for data abstraction within a visual KAT-like environment is to be found in (Reeker, 1986).

4. Some Specific Areas of Tool Application

Suppose that the whole process of creating software for some application like running a robotic assembly line, were developed using a tool like KAT. At the simplest level of application, one could use the tool for recording and refining specifications for the system, on which project assignments could be superimposed. The description of the system could be developed by individual software engineering teams right down to the level of code. Specification, code, and documentation could be kept together, thus simplifying the project librarian's job. Clearly, one could use KAT's facilities for creating graphical AND/OR and flowchart representations of a process to add to the documentation. Further, one might imagine that KAT's Prediction Mode queries such as "What could go wrong here?" and "How could the process be redesigned so as to avoid this problem?", as well as others that could be added, would be helpful in fleshing out program content and making programs "correct by design". Finally, the ability to "step through" a program in KAT's Diagnosis Mode would provide an ideal facility for debugging.

The ways in which a tool such as KAT might be applied to software engineering efforts extend far beyond these specific examples. Below are listed seven generic problem areas of concern to software engineers that we feel could be aided through techniques developed by knowledge engineers.

(1) **Code Design.** This is the primary issue addressed above. Generation of code could take place automatically through specification of processes, just as knowledge bases are automatically generated within KAT.

(2) **Project Design.** In a large software project, the organization of the project reflects, to some level, the organization of the system. How better to determine system segmentation than through gathering the expertise of persons with knowledge of the process for which

the software is being designed? Once this information about the hierarchical project design is obtained, it can be made available to those managing the project who need the "big picture". Given an appropriate knowledge acquisition tool with a knowledge base that can be consulted centrally, managers and domain experts can add comments pertinent to the project design.

(3) **Interteam and Intrateam Communication.** Any sort of computerized tool for project segmentation can facilitate communication within and among programming teams. This is analogous to the sharing of opinions among experts and AI programmers through the use of a knowledge engineering tool such as KAT.

(4) **Problem Diagnosis.** As was mentioned above, a facility such as the Diagnosis Mode in KAT serves an important debugging function, much like an interpreter that allows the programmer to step through a program a statement at a time. Such a feature can provide programmers with an opportunity to observe the consequences of having specified the program flow in a particular way and to embellish or redesign as necessary.

(5) **External Documentation.** Coupling a "software knowledge acquisition system" being used in a software development project to an expert system shell, such as the KEE system used in KAT, would be an effective way of capturing the knowledge needed to develop reference systems that could anticipate and react to a user's needs, rather than producing "canned" documentation. This capability would be useful for many purposes, including tutoring new users and providing intelligent interfaces.

(6) **Redesign and Reuse.** By archiving the knowledge concerning the software design process it should be possible to facilitate the maintenance, redesign and reuse of the modules in a program. The importance of internal documentation to this process has often been remarked. The sophisticated history gathered by KAT should be doubly useful in this

process. It will be necessary to develop tools to access the appropriate knowledge without being saturated with additional irrelevant information, of course. Representational aspects of KAT, including class hierarchies (inheritance) and instance variables tend to promote reuse and redesign. As Fischer (1987) points out, inheritance allows simple redesign by allowing new objects similar to ones previously defined to be created with only a few incremental changes. Inheritance also reduces the need to specify redundant information and simplifies updating and modification. Kaiser and Garlan (1987) discuss the utility of instance variables and associated methods in promoting software reusability. The same information that instance variables provide would be available in such a system.

(7) **Testing and Verification.** As an aid to software engineering it has long been considered useful to provide interpreters that can simulate the execution of code in a production environment, thus helping to test that code. There is also the possibility of verification of the code, either heuristically, with relevant "expertise" built into the software engineering system, or through mathematical techniques such as the inductive assertion method (Floyd, 1967; Hoare, 1969). The "what?" and "why?" questions of KAT elicit much the same information as that in Floyd's suggestions for a verifying compiler.

With respect to the inductive assertion method in particular, there have been a number of efforts over the years at developing machine-aided methods of generating and proving assertions (Floyd, 1972). A KAT-like process knowledge system would provide a good framework for application of these methods. In fact, the current prototype for materials knowledge engineering is designed to gather as much information as possible regarding the expected state of the process at each step of the process during the Prediction Mode, in order to use it during the Diagnosis mode, which is the closest thing that the system now has to verification. With

the projected inclusion of qualitative physical knowledge, KAT would be closer (within its current materials processing domain) to a formal verification of the inductive assertion type, but with the verification being done at a higher level than normally suggested for computer programs. It might actually be more practical to do verification at this level. Though higher level verification would mean separate software/knowledge engineering verification systems for different domains of process control, CAD/CAM, operating systems, numerical algorithms, and other areas, it would be much closer to the way that expert programmers who also understand the content area of the program informally check the functioning of their program. Because of its hierarchical development of program logic, KAT might profitably be coupled with both functional verification and statistical testing (Mills, 1986).

5. Conclusion

The KAT tool is designed for use in knowledge engineering, in the particular task of acquiring expert knowledge about processes. A number of the considerations in its design are derived from software engineering. On the other hand, KAT also provides an excellent potential tool for software engineering. In seeing this, it is useful to consider that programming is really the specification of a process, and that much of what software does today is better conceptualized in this framework than in terms of the traditional problem solving paradigm.

6. References

Biewald, J., P. Goehner, R. Lauber, and H. Schelling (1979). EPOS – A specification and design technique for computer controlled real-time automation systems, *Proceedings of the 4th International Conference on Software Engineering*, Munich, IEEE Computer Society, 245-250.

Blaxton, T. A., and C. R. Westphal (1988). Combining explicit queries with simulation techniques during knowledge acquisition, *Proceedings of the 1988 Eastern Simulation Conference*, Orlando.

Bobrow, D. (1985). *Qualitative Reasoning about Physical Systems*, MIT Press, Cambridge, MA.

Cherry, G. W. (1986). The PAMELA Designer's Handbook, The Analytical Sciences Corporation, Arlington, VA.

Cooke, N. M., and J. E. McDonald (1986). A formal methodology for acquiring and representing expert knowledge, *Proceedings of the IEEE*, 74, 10, 1422-1430.

De Kleer, J. (1984). How circuits work, *Artificial Intelligence*, 24. Reprinted in Bobrow (1984), 205-280.

DuBois, E., J. Hagelstein, E. Lahou, F. Ponsaert, and A. Rifaut (1986). A knowledge representation language for requirements engineering, *Proceedings of the IEEE*, 74, 10, 1431-1444.

Fischer, G. (1987). Cognitive view of reuse and redesign, *IEEE Software*, 4, 4, 60-72.

Floyd, R. W. (1972). Toward interactive design of correct programs. *Information Processing 71: Proceedings of IFIP Congress 1971*, 1, North-Holland, Amsterdam, 7-10.

Floyd, R. W. (1967). Assigning meanings to programs, *Proceedings of the American Mathematical Society Symposia in Applied Mathematics*, 19, 19-31.

Greenspan, S. J., and J. Mylopoulos (1982). Capturing more world knowledge in the requirements specification, *Proceedings of the 6th International Conference on Software Engineering*, Tokyo, 225-234.

Hamilton, M., and S. Zeldin (1976). Higher-order software — a methodology for defining software, *IEEE Trans. Software Engineering*, SE-2, 1, 9-32.

Harel, David (1980). AND/OR programs: A new approach to structured programming, *ACM Transactions in Programming Languages and Systems*, 2, 1, 1-17.

Hoare, C. A. R. (1969). An axiomatic approach to computer programming, *Communications of the ACM*, 12, 10, 576-580.

IntelliCorp (1986). *The SimKit System: Knowledge-Based Simulation Tools in KEE*, Document No. 1.1-USK-1, IntelliCorp, Mountain View, CA.

Jackson, M. A. (1975). *Principles of Program Design*, Academic Press, New York.

Kaiser, G. E., and D. Garlan (1987). Melding software systems from reusable building blocks, *IEEE Software*, 4, 4, 17-24.

Kuipers, B. (1984). Commonsense reasoning about causality: Deriving behavior from structure, *Artificial Intelligence*, 24. Reprinted in (Bobrow, 1985), 169-204.

Mills, H. D. (1986). Structured programming: Retrospect and prospect, *IEEE Software*, 3, 6, 58-66.

Page-Jones, M. (1980). *The Practical Guide to Structured Systems Design*, Yourdon Press, New York.

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15, 2, 1053-1058.

Reeker, L. H. (1986). Extended AND/OR graphs and parallel programming in a graphic environment, *Proceedings of International Computing Symposium 1986*, Tainan, Taiwan, December 1986, 138-146.

Roediger, H. L., and T. A. Blaxton (1987). Retrieval modes produce dissociations in memory for surface information. In *Memory and Cognitive Processes: The Ebbinghaus Centennial Conference*, ed. D. S. Gorfein and R. R. Hoffman, Erlbaum, Hillsdale, NJ.

Schacter, D., and P. Graf (1986). Effects of elaborative processing on implicit and explicit memory for new associations. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 12, 432-44.

Suwa, M., A. C. Scott, and E. H. Shortliffe (1982). An approach to verifying completeness and consistency in a rule-based expert system. *The AI Magazine*, 3, 3, 16-21.

Weinberg, G. M. (1971). *The Psychology of Computer Programming*, Van Nostrand Reinhold, New York.

Westphal, C. R. (1988). Using simulation-like methodologies for knowledge base validation. Submitted to the 3rd International Symposium on Knowledge Engineering, Madrid.

Winograd, T. (1979). Beyond programming languages. *Communications of the ACM*, 22, 7, 391-401.

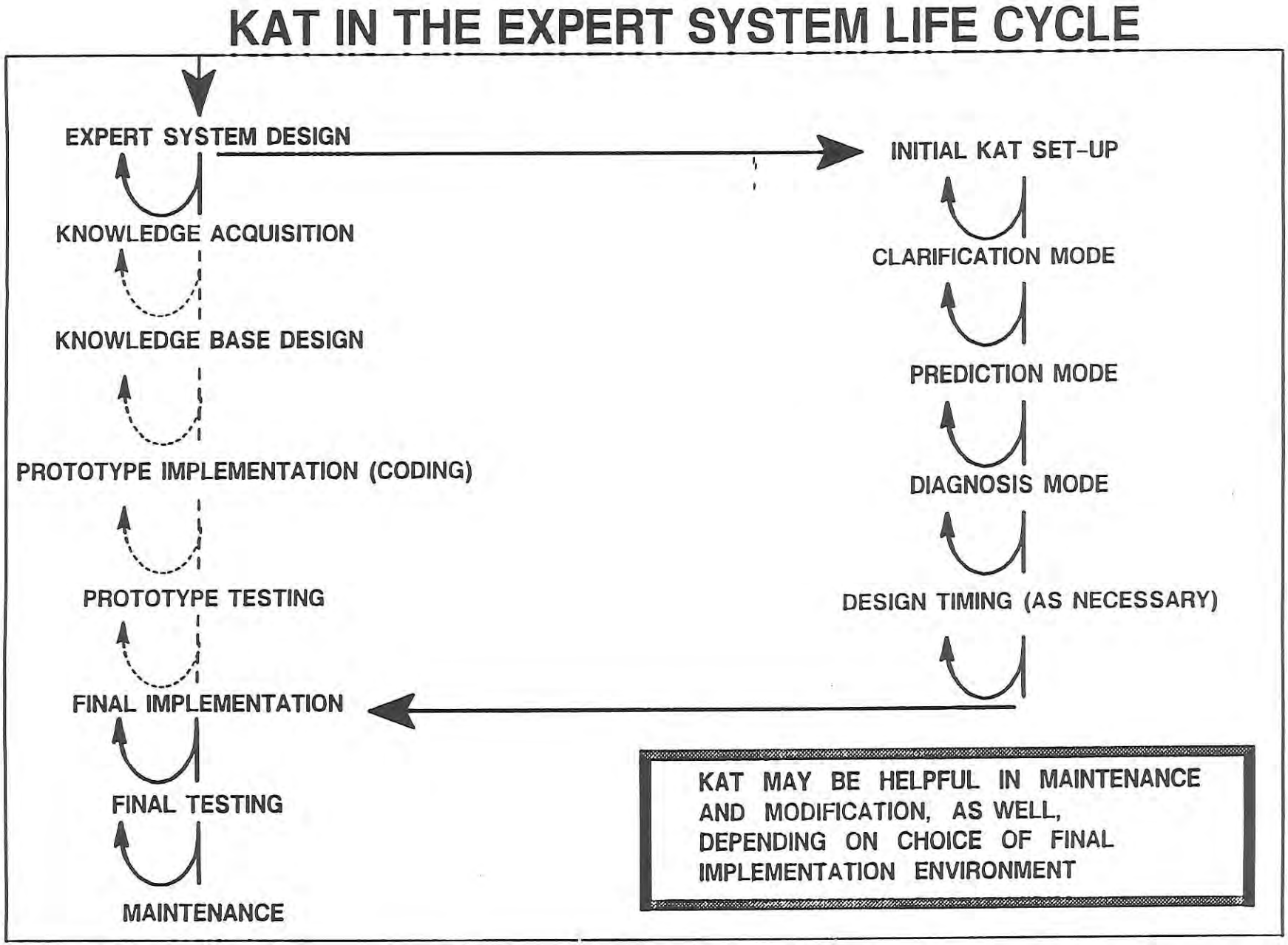
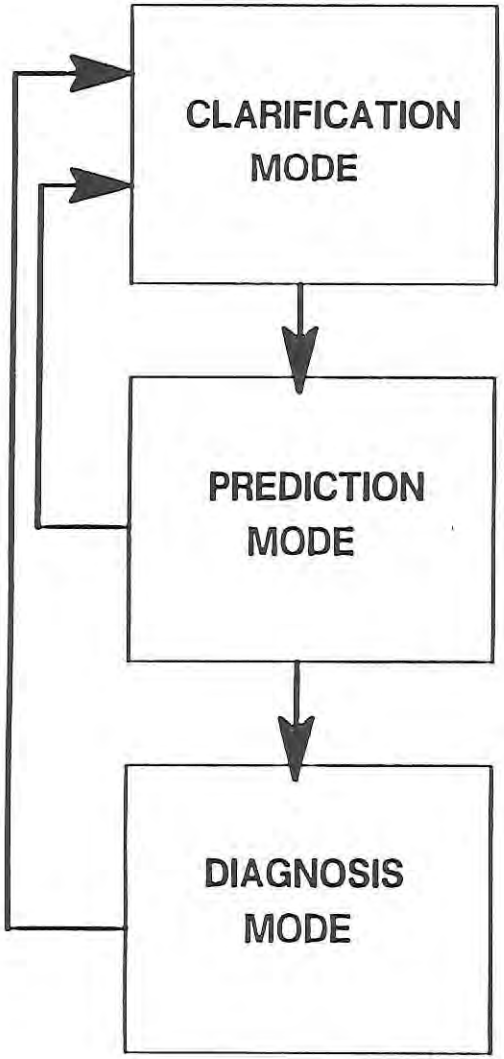

KAT IN THE EXPERT SYSTEM LIFE CYCLE
EXPERT SYSTEM DESIGN
KNOWLEDGE ACQUISITION
KNOWLEDGE BASE DESIGN
PROTOTYPE IMPLEMENTATION (CODING)
PROTOTYPE TESTING
FINAL IMPLEMENTATION
FINAL TESTING
MAINTENANCE
INITIAL KAT SET-UP
CLARIFICATION MODE
PREDICTION MODE
DIAGNOSIS MODE
DESIGN TIMING (AS NECESSARY)
KAT MAY BE HELPFUL IN MAINTENANCE AND MODIFICATION, AS WELL, DEPENDING ON CHOICE OF FINAL IMPLEMENTATION ENVIRONMENT

Figure 1

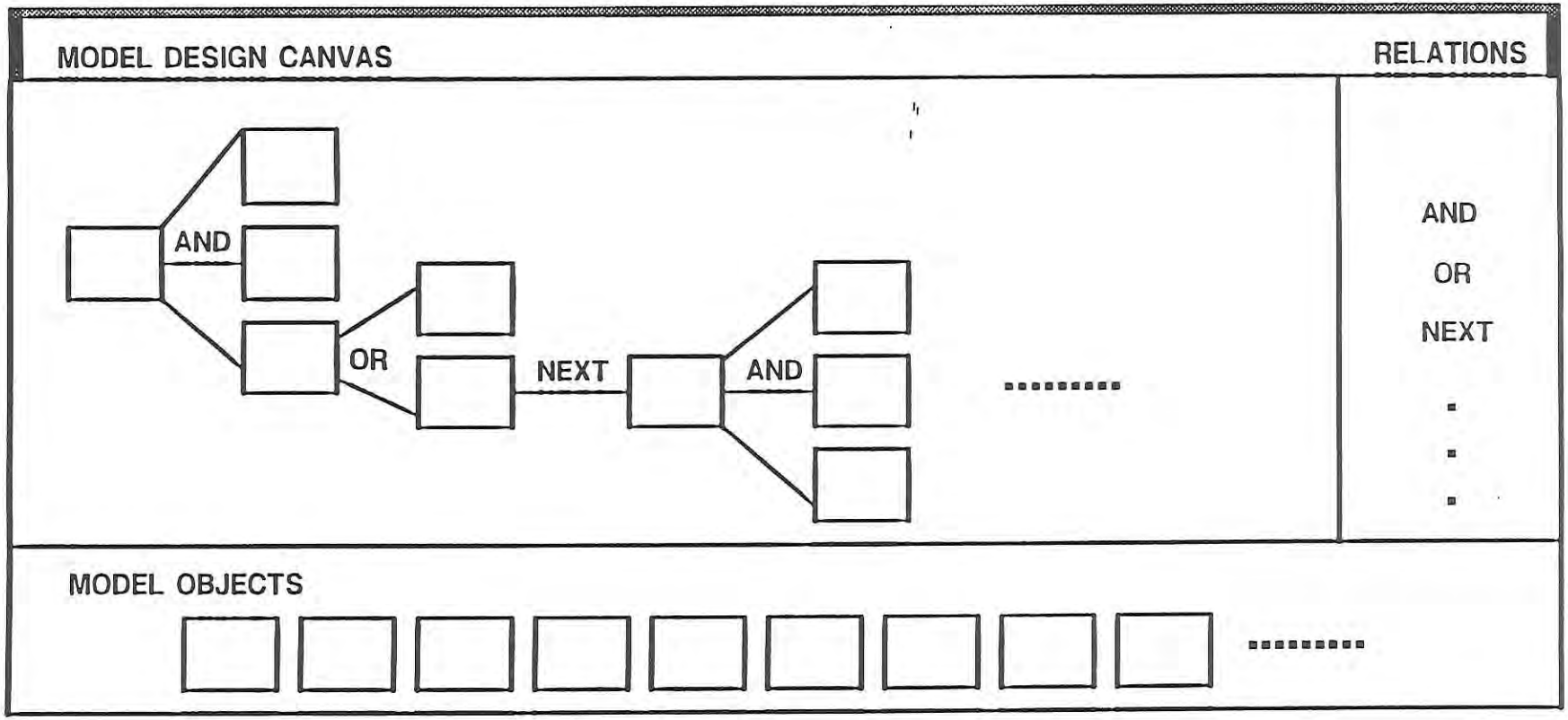
THE OVERALL STRUCTURE OF KA



- Description of process decomposition is created in the form of an AND / OR graph.
- Logical relationships among processes are represented explicitly.
- Information is presented in flowchart format, and expert checks
  - inputs & outputs
  - relevant attributes
  - process rules
- Information format is changed to show temporal ordering; procedural knowledge elicited.
- Dynamically step through knowledge structure, making changes as necessary.
- Implicit interactions within the process model are explicated.

Figure 2

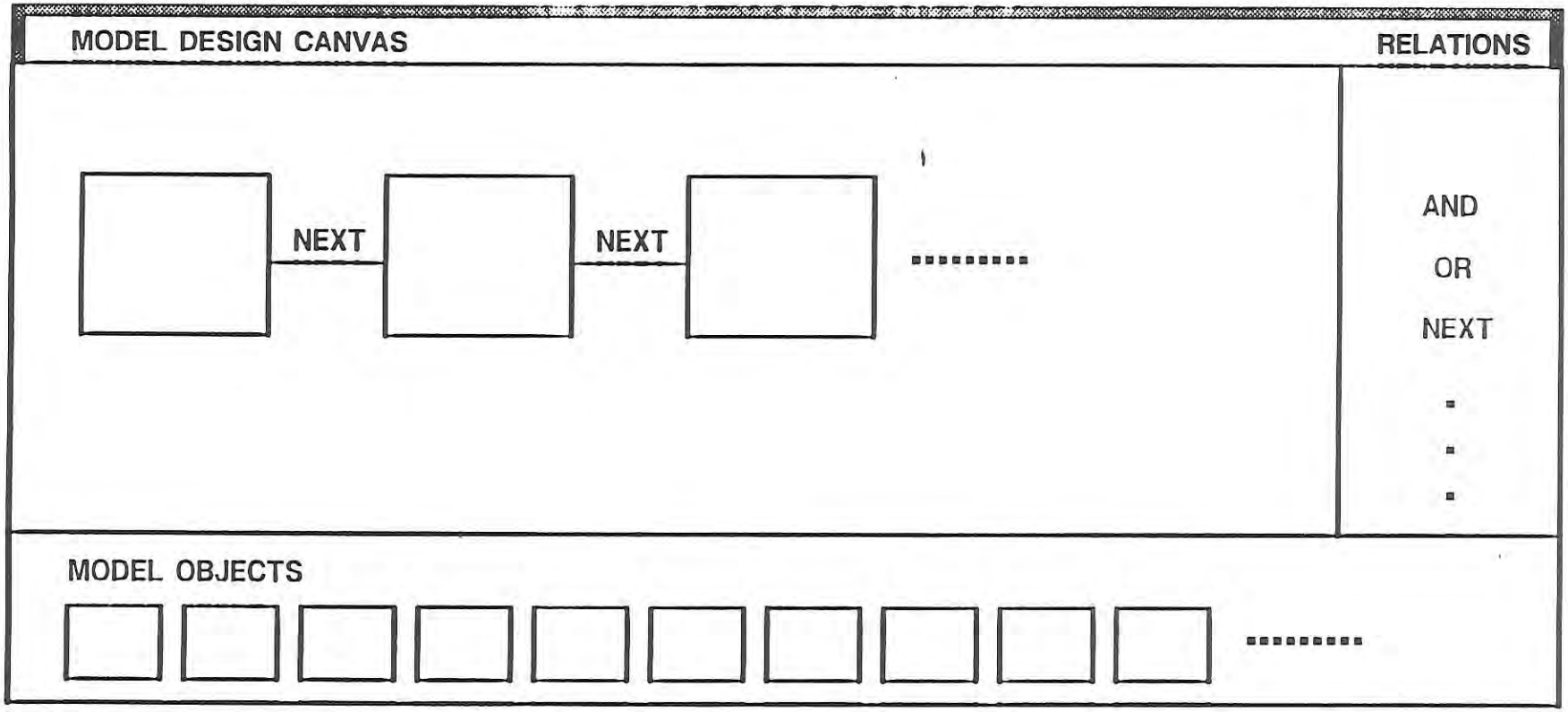
CLARIFICATION MODE



- Expert starts from scratch and decomposes process into a logical progression of subprocesses.
- Alternative subprocesses are indicated where appropriate (ORs).
- Result is an AND/OR graph of the entire process.

Figure 3

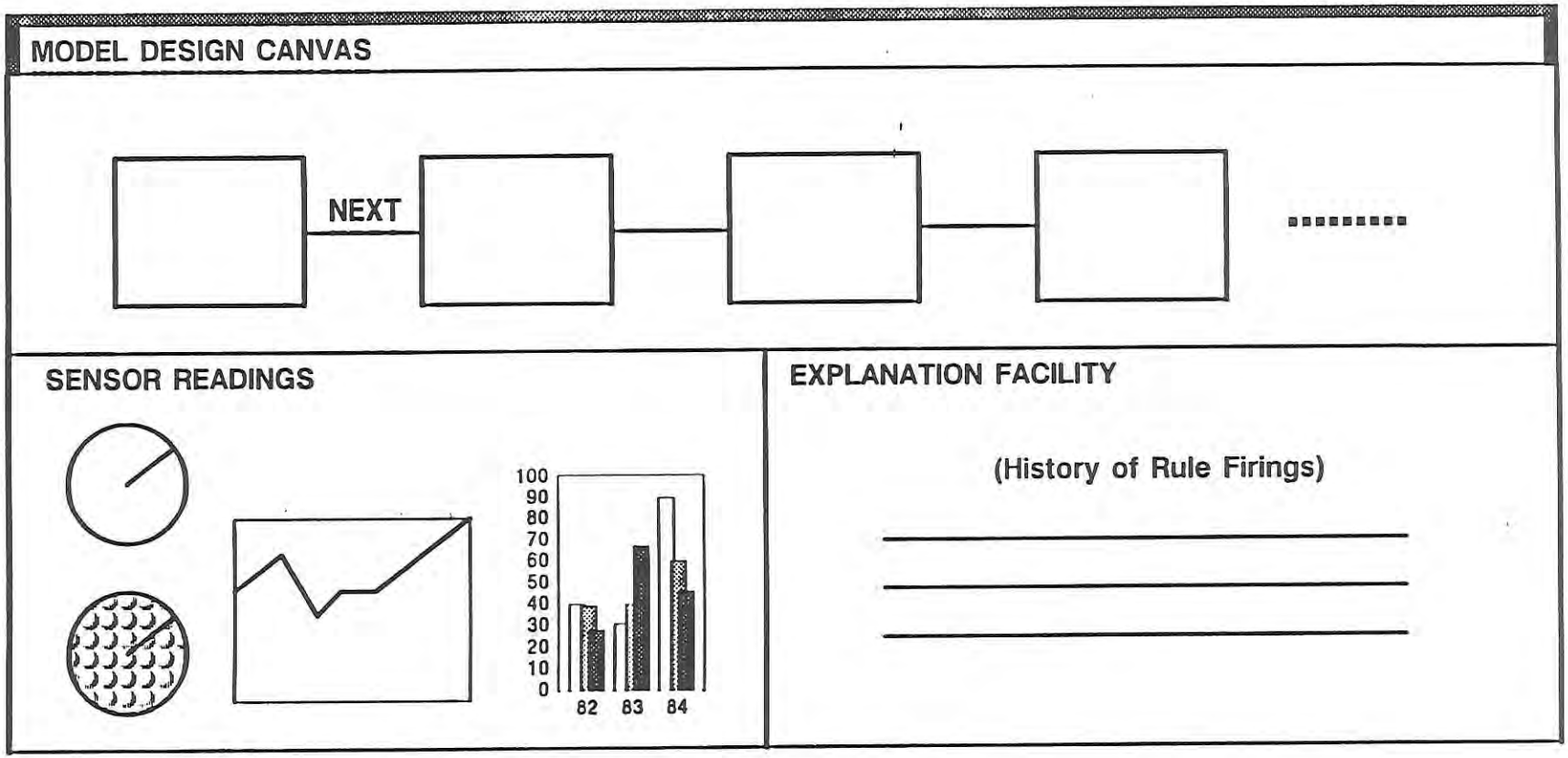
PREDICTION MODE



- KNOWLEDGE STRUCTURE CREATED IN THE CLARIFICATION MODE IS REDISPLAYED IN A TEMPORAL (FLOWCHART) FORM.
- FOR EACH SUBPROCESS, THE EXPERT IDENTIFIES:
  - INPUTS AND OUTPUTS
  - OTHER RELEVANT DATA STRUCTURES
  - RULES WHICH DEFINE RELATIONSHIPS AMONG SUBPROCESSES, ETC.
- TEMPORAL ORDERING MAY BE EASILY MODIFIED

Figure 4

DIAGNOSIS MODE



- KNOWLEDGE STRUCTURE IS PRESENTED IN FLOWCHART FORM.
- EXPERT SEES A DYNAMIC STEP-THROUGH DISPLAY OF THE PROCESS WITH RELEVANT SENSOR READINGS AND THE HISTORY OF RULE APPLICATIONS.
- THIS STEP-THROUGH SERVES AS AN IMPLICIT QUERY, PROMPTING THE EXPERT TO REDESIGN UPON NOTICING INCOMPLETE OR FAULTY SPECIFICATIONS.

Figure 5

Using Simulation-like Methodologies for Knowledge Base Validation

Christopher R. Westphal

The BDM Corporation, Advanced Technology Group
7915 Jones Branch Drive, McLean, VA 22102

Abstract

A primary factor for establishing the validity of an expert system depends on how well the system's knowledge corresponds with the real world environment being modeled. These correspondences (knowledge) may be defined as components, such as heuristic methods, object representations, and relational dependencies. This paper will discuss several validation procedures, in particular simulation-like methodologies, which allow a dynamic overview of the knowledge structure to be presented as the interactions of its related components. In addition, a knowledge engineering tool (prototype) will be discussed and related to these validation procedures in an attempt to express future areas of innovation.

1. Introduction.

1.1. Simulation for Validation.

As today's expert systems become larger and more complex there must be better means of validating the correctness of their knowledge structures and rule bases. One plausible method for doing this is to provide a diagnostic analysis of these structures using simulation-like methodologies to insure the structures correctly represent the real world environment being modeled. Since the most critical stage of expert system development occurs during the elicitation and transfer of domain knowledge from the expert into machine representations, it appears reasonable to perform extensive testing and validation at this point.[1]

Knowledge base validation is an important aspect of knowledge acquisition. The validity of a system may be determined directly from the expected performance

[1] The distinction between testing and validation, as defined by Muchnick [1981], states that testing involves the effort of searching for errors and validation is used to confirm the absence of errors.

intended from a specific set of input parameters. Although there exists no standard set of procedures for establishing the credibility of a system based on performance, many environments execute some form of static overview of the system. The way a system manages its internal components (ie. rules, objects, relations) determines how the system will perform. These components are responsible for representing the elements of the "real system" and must be accurate if the system is to operate correctly. Non-static evaluation techniques[2] using simulation allow the behavior of a system to be viewed as interactions among the internal objects of a model [Payne, 1982] and can be incrementally applied in phases to the system allowing the model to be tested for the accurate representation of the relevant features of the system being developed [Payne, 1982] [Mihram, 1973].

1.2. Phased Sensitivity Analysis.

An ideal model would be expected to correspond *exactly* with a real world system in its relevant input-output behavior. Although this can never be fully achieved, a close approximation of a system may be validated through careful observations of the model structure using sensitivity analysis.

Basic sensitivity analysis techniques view the model structure as an input-output process (similar to a black-box approach) providing feed-back about incremental changes of the input with relation to the output. The input parameters represent the range of operational values for the particular process they are associated with. As these input values are altered, corresponding perturbations of certain output variables will determine if the model correctly defines the intended system. Any changes in output variables not directly associated with any of the input variables will be suspected of

[2] Dynamic validation techniques not applicable to an incremental design are not discussed here.

misrepresentation.[3] In addition, inflated changes in the output relative to nominal variations of input parameters may indicate an improper reasoning technique in the model [Miller, 1974].

A system may be decomposed into stages (ie. subsystems or phases) [Yourdon, 1979], allowing an extension of sensitivity analysis to be applied to each stage during development. Thus phased sensitivity analysis permits the outputs of a discrete stage to be observed as functions of the related input variables. This provides a localized error detection scheme for validating the current stage. Since each stage has been isolated from the rest of the system it contains less complex interactions to complicate the validation procedures and makes the project development more manageable [O'Keefe, 1987]. However, successful validation of all stages does not necessarily imply the system as a whole will be valid, therefore a sensitivity analysis should also be applied to the entire structure after each stage has been confirmed.

1.3. Multiple Experts.

The need for additional review procedures may provide useful support when establishing the credibility of the model. The usage of multiple experts for validation of a knowledge structure provides an excellent method to insure the structure is complete and contains minimized biased information. Commonly, each expert is a specialist about a particular aspect of the production domain [Mittal, 1985] [Cochran, 1987] and will build upon the existing base structure, adding their own knowledge as necessary.

[3] Analysis of multivariate-response may prove difficult during a simulation due to the interdependencies between variables [Friedman, 1984].

When multiple experts are used to produce a knowledge structure, all relevant information must be presented in a format which can be easily comprehended by all the participants involved [Payne, 1982]. However, conflicts may arise when there are differences in opinions about how information should be presented. These may be due to variances in terminology or structural disagreements [Wielinga, 1985], and should be resolved appropriately. Presenting the material in an organized and effective manner may suppress certain initial interpretational conflicts between expert viewpoints [Payne, 1982], but may provide no insight on the interaction effects of another expert's opinions. Using a simulation-like presentation of the model will allow implicit multi-expert conflicts to be made explicit. Many of these conflicts may be resolved by simulating multiple opinions (methods) in-order to exhibit similar behavior in their functionality. This similarity can be used to persuade an expert to accept an alternative means for producing the desired solution, and thus removing repetitive information from the knowledge structure. If a disagreement is still evident, the system should accommodate both representations and provide facilities to select the preferred method of reasoning [Cochran, 1987].

The appropriate representations for supporting multiple experts would modify the early conceptual design stages of a system to include facets labeled with the appropriate expert who provided the information [Reeker, 1986]. This allows future experts to record their own opinions without losing the base information already acquired.

2. Knowledge Acquisition Tool.

The facilities for obtaining the base structure of a model should be carefully integrated with the validation procedures of the system, as to insure an accurate representation of the required specifications. A system currently under development at

The BDM Corporation is the Knowledge Acquisition Tool (KAT), which provides both a medium for acquiring specialized domain knowledge in materials processing and a structured presentation of the model to interactively validate system correctness.

KAT has been developed as a tool to be used in conjunction with a knowledge engineer to help expedite the knowledge acquisition process. Hybrid systems such as KRITON [Diederich, 1987], SALT [Marcus, 1985], and INFORM [Moore, 1987] try to automate the elicitation process (via removal of the knowledge engineer) so the expert can interact directly with the system. This can degrade the system efficiency through misrepresentations of information and underutilized reasoning techniques. KAT has been automated to a certain degree but is not envisioned as a replacement for the knowledge engineer [Blaxton, 1988]. Although KAT uses several conceptual models, they have been carefully integrated and matched to work with the pre-selected domain (as do KNACK [Klinker, 1987] and OPAL [Musen, 1987]). More general purpose systems such as MOLE [Eshelman, 1986], KADS [Hayward, 1987], and MORE [Kahn, 1985] contain no specific subject knowledge and can be applied to broad domain applications which may restrict the integrity of the knowledge base. Many systems provide some form of generic verification and testing (i.e. constraint violations and syntax errors), but the system validation procedures (if any) are more dependent on the design environment used to produce a system [Hoover, 1984].

Sayre [1963] advocates a design methodology for a system model based on replication, formalization and simulation. KAT has been developed using a modified interpretation of these guidelines. The user interacts with KAT in three query modes (Figure 1), each designed to elicit different forms of knowledge. The Clarification Mode (replication) captures primary declarative knowledge through the construction of

an AND/OR model. The Prediction Mode (formalization) converts the model into a process flow representation so it can extract procedural knowledge, and the Diagnosis Mode (simulation) provides a step-through presentation of the model to reveal any observable design faults. Each of these modes is an essential link to the construction of a KAT knowledge base.

Although the first two modes are necessary for the verification of a knowledge base and include facilities for completeness checks (such as unreferenced and illegal attribute values, syntax checking, and minor inconsistencies) [Nguyen, 1987], the last mode is of primary importance to validate the correctness of the working model. Potentially serious misconstructions include consistency checks for redundant, conflicting, and circular rules [Nguyen, 1987][Suwa, 1982], in addition to unreachable conclusions and misconstrued logical relationships [Hayes- Roth, 1985]. To effectively construct valid rule premises entails subjective reasoning about state changes and insight of the execution effects [Hayes-Roth, 1985][Williams, 1983], a task not easily accomplished. However, KAT can monitor a process execution by graphically presenting the internal interactions between rules, objects and attributes. A more in-depth description of KAT follows.

3. Overview of KAT.

3.1. The Clarification Mode.

The Clarification Mode (Figure 2) allows the domain expert to build-up a representation of a process structure forming a replication of the system to be modeled. The model objects represent real world items in the materials processing environment (ie. sensors, reactors, ceramics, etc). The attributes of an object define

its physical characteristics and the relationships between them correspond to their formal activities [Sayre, 1963]. The interactions of multiple experts are supported by providing "guard" operators affixed to the relations that are labeled with the expert who provided the information [Reeker, 1986]. Adopting this type of flexible knowledge structure provides alternative reasoning techniques for system operations [Cochran, 1987]. An object library (Figure 2.a) furnishes a finite number of relevant entities pertaining to the domain subject, allowing the expert to construct the model structure [Blaxton, 1988]. An option to create additional objects through a simplified user interface is also provided. Objects are placed in a design viewport and relations are asserted between them thus forming the canonical structure to be used through out the system development.

3.2. The Prediction Mode.

The Prediction Mode (Figure 3) converts the basal structure into a flowchart format to provide an alternative view in an attempt to elicit additional information where it is incomplete [Reeker, 1987]. The structure is reduced to composite objects using an equivalent node aggregation scheme [Reddy, 1986], thus simplifying the structural format for analyses by suppressing the intermediate details of the model [Martin, 1968][Reddy, 1986]. The input/output parameters of the composite structures will be defined at this stage, thus obtaining some of the fundamental information required to perform a sensitivity analysis of the model [Tomovic, 1972] [Payne, 1982]. In addition, queries for sensor production are constructed with active images portraying the current values of the respected unit. The qualitative information elicited about these sensors is then obtained through specification of rule constructs. Due to the difficulties involved with eliciting procedural knowledge [Cooke, 1986] it is expected that these

two modes (Prediction, Clarification) will be iterated numerous times before a solid knowledge structure is defined [Reeker, 1987][De Graef, 1985][Nguyen, 1987].

3.3. The Diagnosis Mode.

The Diagnosis Mode (Figure 4) uses a discrete-event scheduling method where the rule actions determine the system's behavior. The initial specification may be preliminary, thus exposing underdeveloped sections of the model, but as additional knowledge is obtained, it will become increasingly explicit [Reeker, 1987]. Since the model uses objects as input/output processes, a model inference [Mihram, 1973] or sensitivity analysis can be applied to the model. Using a sensitivity analysis to provide a structured presentation of the domain model will allow the expert to insure a proper representation of the system has been implemented based on the face validity of the model [Hermann, 1967], in addition, variability estimates via multiple process simulations may also be obtained [Hayes- Roth, 1983]. The parameters of the inference are specified using an interactive inspectable interface. This type of interface allows experts to interact with system activities that cannot be normally (ie. cost effective) be monitored or manipulated in a real world environment [Hollon, 1984]. Relevant sensors and value displays will emulate the status of the respected model components (ie. flow rates, fluid levels, switch positions, etc) and manipulation of these displays (via mouse) will allow the expert to interactively adjust their values. The analysis of discrete intervals for a range of values may then be compared to expose differences in the resultant states created, which may help express alternative parameters for system operations [Baskaran, 1984] [Miller, 1974]. If the analysis exposes a need to redesign parts of the model structure, the expert can return to the previous modes (Clarification and Prediction) and make the changes required.

4. Simulation Environment.

KAT is being prototyped on a Symbolics 3670 machine supported by KEE and SIMKIT environments.[4] The SIMKIT environment provides numerous facilities (clock/calendar, verification procedures, data collectors, and event generators) for project development [IntelliCorp, 1986]. The clock used to represent the current simulation time should be associated with the calendar to schedule future events and maintain priorities [Reddy, 1986][Franta, 1977]. This combination will support run time checkpoints for the model that allow step through simulation (via the clock), and predetermined stop points (via the calendar). A hierarchical design for the model development was desired to promote inheritance and provide general structure verification procedures for constraint violations which include basic checks for cardinality faults, incompatible value classes, and undefined methods [IntelliCorp, 1986]. Mechanisms for data collection (i.e. dynamic, static, stochastic) are used to review, compare, and analyze the activities of the sample simulations from the data produced by event generators [Martin, 1968]. These generators introduce variability into simulation through generation of deterministic, pseudo-random, and arbitrary sequences of test data.

5. Conclusion.

A brief overview of the KAT system has been presented to show our current efforts designing a knowledge acquisition system for materials processing. More importantly, this system has linked the usage of simulation- like methodologies to knowledge base validation.

[4] KEE and SIMKIT are trademarks of IntelliCorp.

One particular simulation method of importance is sensitivity analysis. Variations of causal input parameters are used to establish an acceptance of the model through observed changes in relative output variables. This may be used to insure a proper representation of the real system has been correctly implemented.

As of today, few A.I. systems have used this functionality to validate their knowledge structures. However, KAT has been designed with facilities to produce a structured presentation of its knowledge base (ie. sensitivity analysis), coupled with a flexible interface to allow the expert to interactively adjust parameter values, thus providing a means to analyze changes in the state of the model.

As expert system technology progresses, more emphasis will be placed on developmental sensitivity analysis (as opposed to field testing); allowing the expert to analyze his/her opinions as they are developed and view how the overall system is directly affected by them, thus producing more reliable knowledge bases.

Acknowledgements

Special thanks to Drs. Larry Reeker, Teresa Blaxton, and Roger Geesey.

References.

Baskaran, V., Y.V. Reddy (1984). "An Introspective Environment for Knowledge Based Simulation," *In Proceedings of the 1984 Winter Simulation Conference,* pp. 645-651.

Blaxton, T. A., C. R. Westphal, (1988). "Combining Explicit Queries with Simulation Techniques During Knowledge Acquisition," *To appear in Proceedings of Eastern Simulation Conference,* Orlando, FL, 6 pages.

Cochran, E. L., B. L. Hutchins (1987). "Testing, Verifying, and Releasing an Expert System: The Case History of Mentor," *In Proceedings of Third Conference on Artificial Intelligence Applications,* Kissimmee, FL, IEEE Computer Society, pp 163-167.

Cooke N. M., and J. E. McDonald (1986). "A Formal Methodology for Acquiring and Representing Expert Knowledge," *Proceedings of the IEEE,* Vol. 74, No. 10, pp. 1422-1430.

De Graef, P., J. Breuker (1985). "A Case Study in Structured Knowledge Acquisition," *In Proceedings of Ninth International Conference on Artificial Intelligence.* Los

Angeles, CA, pp. 390-392.

Diederich, J. (1987). "Knowledge-Based Knowledge Elicitation," *In Proceedings of Tenth International Conference on Artificial Intelligence.* Milan, Italy, pp 201-204.

Eshelman, L. and J. McDermott (1986). "MOLE: A Knowledge Acquisition Tool that uses its Head," *In Proceedings of the 5th National Conference on Artificial Intelligence,* Philadelphia, PA, pp. 950-955.

Franta, W. R. (1977). *The Process View of Simulation,* North Holland, New York, pp. 1-26.

Friedman, L. W. (1984). "Multivariate Simulation Output Analysis: Past, Present, Future," *Proceedings of the 1984 Winter Simulation Conference,* pp. 277-280.

Hayes-Roth, F. (1985). "Rule-Based Systems," *Communications of the ACM,* Vol. 28, No. 9, pp. 921-932.

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

Hayward, S. A., B. J. Wielinga, J. A. Breuker, (1987). "Structured Analysis of Knowledge," *Int. J. Man-Machine Studies,* Vol. 26, No. 4, pp. 487-498.

Hermann, C. H. (1967). "Validation Problems in Games and Simulation," *Behavioral Science,* Vol. 12, pp. 216-231.

Hollan, J. D., E. L. Hutchins, L. Weitzman (1984). "STEAMER: An Interactive Inspectable Simulation-Based Training System," *AI Magazine,* Vol. 5, No. 2, pp. 15-27.

Hoover, S. V. R. F. Perry (1984). "Validation of Simulation Models: The Weak/Missing Link," *Proceedings of the 1984 Winter Simulation Conference,* pp 293-295.

IntelliCorp (1986). *The SIMKIT System: Knowledge-Based Simulation Tools in KEE,* Document No. 1-1-USK-1, IntelliCorp, Mountain View, CA.

Kahn, G., S. Nowlan, J. McDermott (1985). "MORE: An Intelligent Knowledge Acquisition Tool," *In Proceedings of Ninth International Conference on Artificial Intelligence.*
Los Angeles, CA, pp. 582-584.

Klinker, G., B. Casey, S. Genetet, J. McDermott (1987). "A KNACK for Knowledge Acquisition," *In Proceedings of the 6th National Conference on Artificial Intelligence,* Seattle, WA, pp 488-493.

Marcus, S., J. McDermott, T. Wang (1985). "Knowledge Acquisition for Constructive Systems," *In Proceedings of Ninth International Conference on Artificial Intelligence.* Los Angeles, CA, pp. 637-639.

Martin, F. F. (1968). *Computer Modeling and Simulation,* John Wiley and Sons, New York.

Mihram, G. A. (1973). "Some Practical Aspects of the Verification and Validation of Simulation Models," *Operations Research Quarterly,* Vol. 23, No. 1, pp. 17-29.

Miller, D. R. (1974). "Sensitivity Analysis and Validation of Simulation Models," *Journal of Theoretical Biology,* Vol. 48, pp. 345-360.

Mittal, S., C. L. Dym (1985). "Knowledge Acquisition from Multiple Experts," *AI Magazine,* Vol. 6, No. 2, pp. 32-36.

Moore, E. A., A. M. Agogino, (1987). "INFORM: An Architecture for Expert-Directed Knowledge Acquisition," *Int. J. Man-Machine Studies*, Vol. 26, No. 2, pp. 213-230.

Muchnick, S. S., N. D. Jones (1981). *Program Flow Analysis: Theory and Applications*, Prentice-Hall, Englewood Cliffs, NJ.

Musen, M. A., L. M. Fagan, D. M. Combs, E. H. Shortliffe (1987). "Use of a Domain Model to Drive an Interactive Knowledge-Editing Tool," *Int. J. Man-Machine Studies*, Vol. 26, No. 1, pp. 105-121.

Nguyen, T. A., W. A. Perkins, T. J. Laffey, D. Pecora (1987). "Knowledge Base Verification," *AI Magazine*, Vol. 8, No. 2, pp. 69-75.

O'Keefe, R., B. Osman, E. Smith (1987). "Validating Expert System Performance," *IEEE Expert*, Vol. 2, No. 4, pp 81-89.

Payne, J. A. (1982). *Introduction to Simulation: Programming Techniques and Methods of Analysis*, McGraw-Hill, New York.

Reddy, Y. V., M. S. Fox, N. Husain (1986). "The Knowledge-Based Simulation System," *IEEE Software*, Vol. 3, No. 2, pp. 26-37.

Reeker, L. H., (1986). "Extended AND/OR Graphs and Parallel Programming in a Graphic Environment," BDM Internal Technical Report BDM/MCL-86-1686-WP, 20 pages.

Reeker, L. H., T. A. Blaxton, R. A. Geesey (1987). "A Knowledge Acquisition Tool for Processes," *In Proceedings of the 5th Intelligence Community AI Symposium*, Washington, D.C., 11 pages.

Sayre, K. M., and F.J. Crosson (eds) (1963). *The Modeling of Mind: Computers and Intelligence*, Notre Dame Press, Notre Dame, Indiana.

Suwa, M., A. C. Scott, E. H. Shortliffe (1982). "An Approach to Verifying Completeness and Consistency in a Rule-Based Expert System," *AI Magazine*, Vol. 3, No. 2, pp. 16-21.

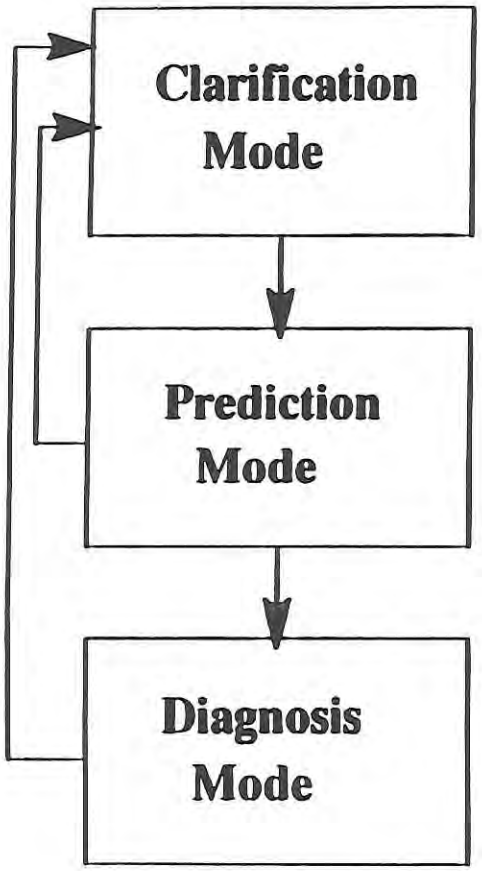
Tomovic, R., and M. Vukobratovic (1972). *General Sensitivity Theory*, American-Elsevier, New York, pp. 1-15.

Wielinga B. J. and J. A. Breuker (1985). "Interpretation of Verbal Data for Knowledge Acquisition,"In *Advances in Artificial Intelligence*,T.

Williams, M. D., J. D. Hollan, A. L. Stevens (1983). "Human Reasoning About A Simple Physical System,"In *Mental Models*, D. Genter and A. Stevens (eds), Hillsdale, N.J., Erlbaum, pp 131-153.

Yourdon, E., L. L. Constantine (1979). *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*, Prentice-Hall, Englewood Cliffs, NJ.

THE OVERALL STRUCTURE OF KAT



+ BASIC LAYOUT (Verification)
  - Construction of AND/OR model
  - Process decomposition
  - Multiple expert support

+ PROCESS FLOW (Verification)
  - Sensor production
  - Rule extraction
  - Active image selection

+ STRUCTURED PRESENTATION (Validation)
  - Interactive/inspectable interface
  - Multiple-value reasoning methods
  - Sensitivity analysis

Figure 1.

KAT INTERFACE FOR CLARIFICATION MODE

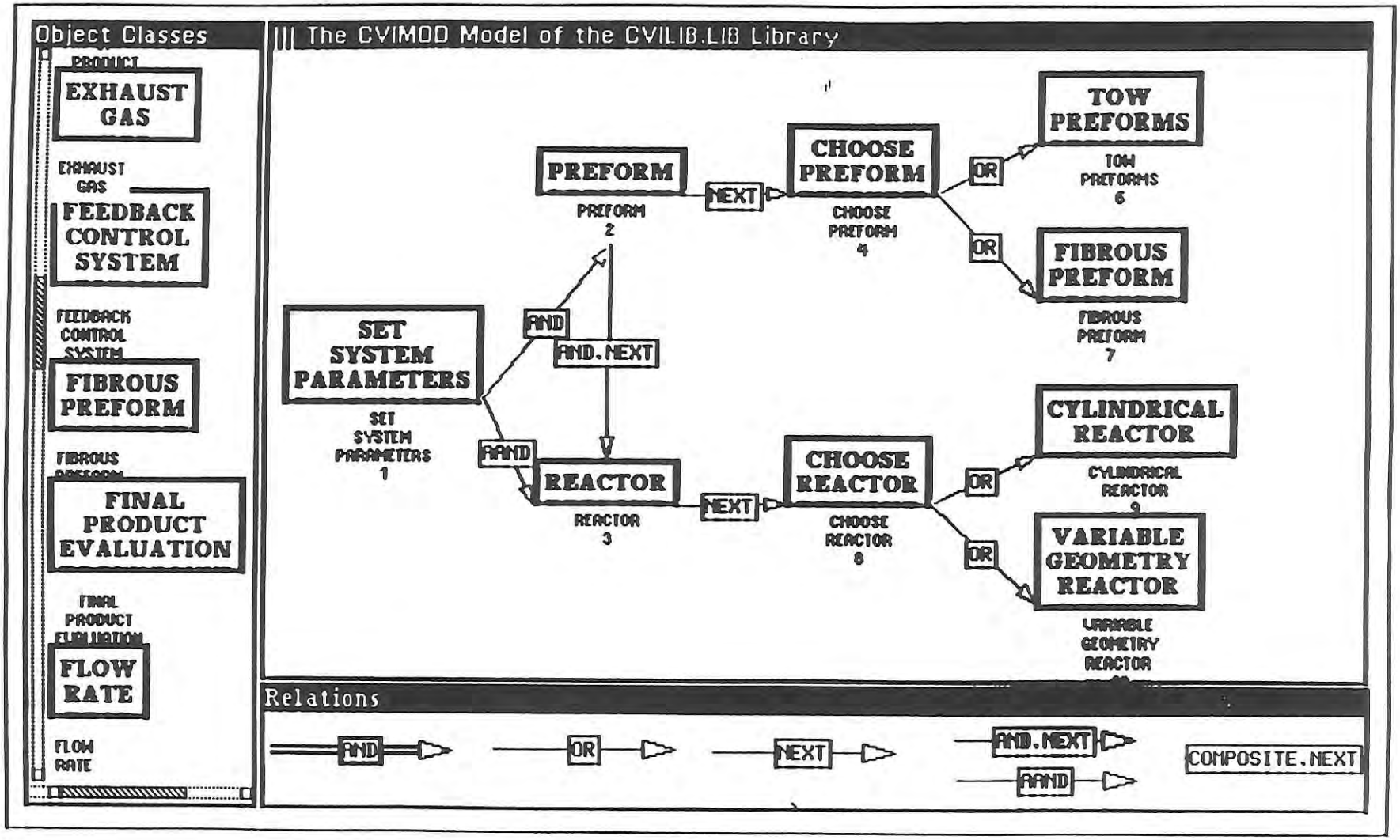


Figure 2.

CVI LIBRARY

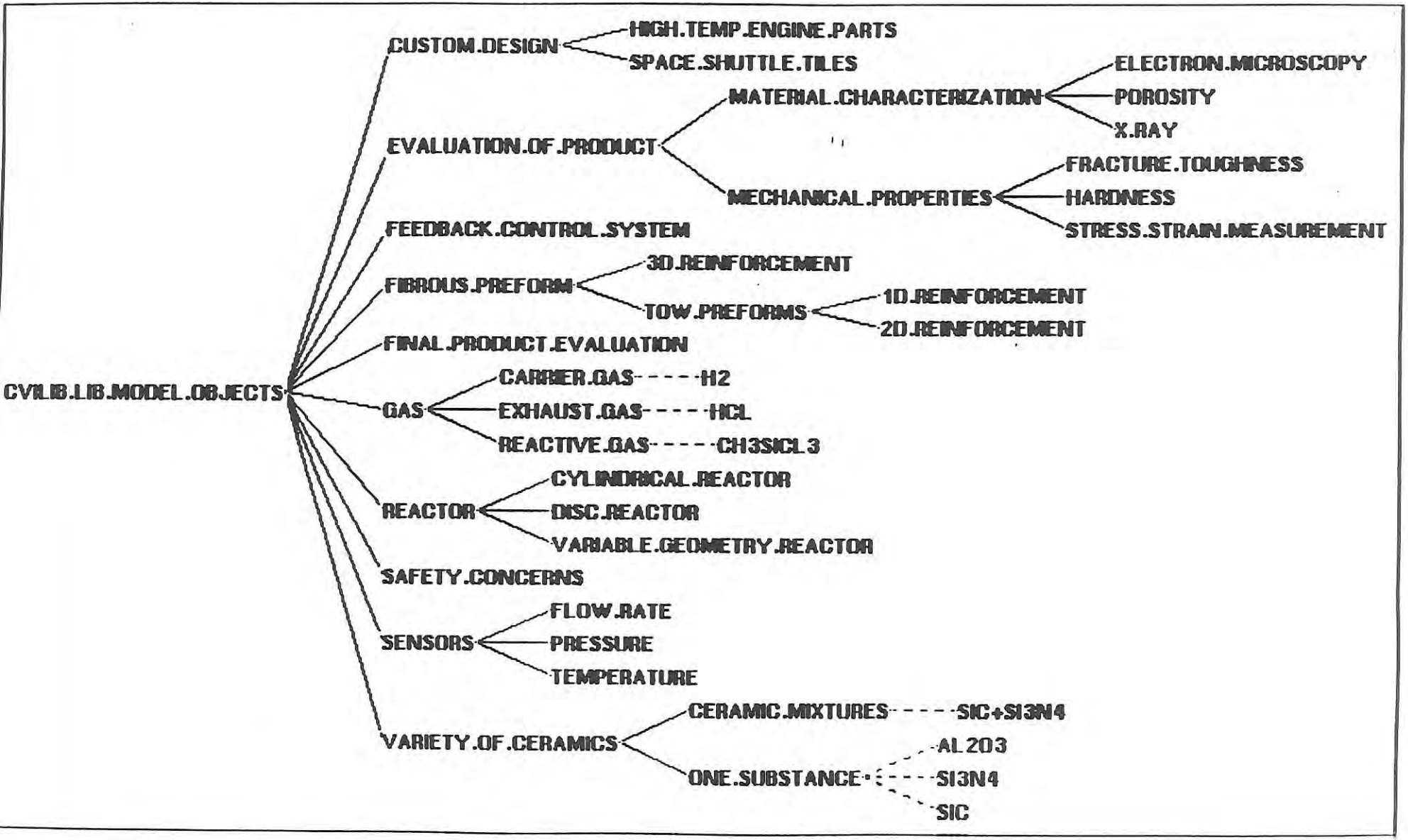
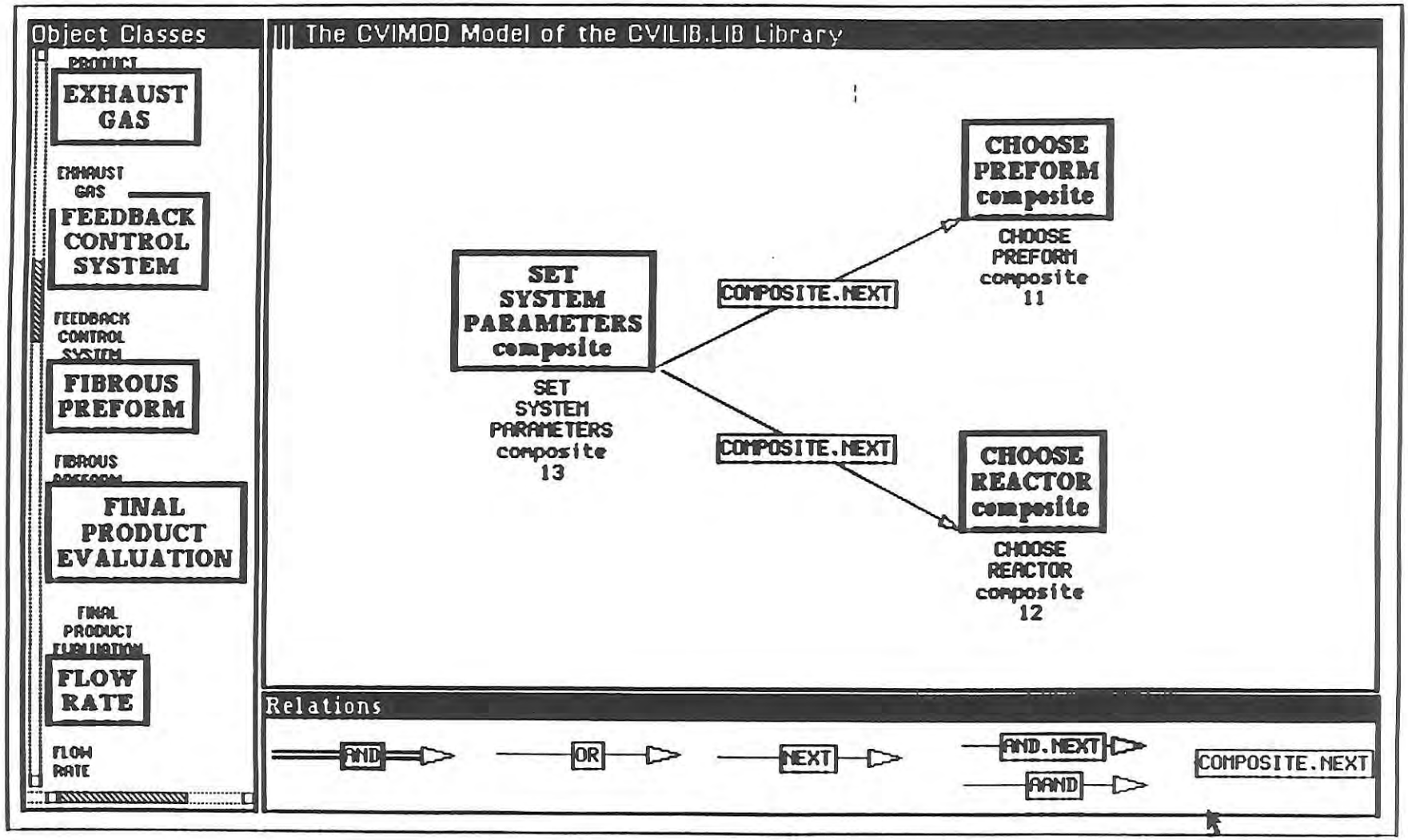


Figure 2.a

KAT INTERFACE FOR PREDICTION MODE



KAT INTERFACE FOR DIAGNOSIS MODE

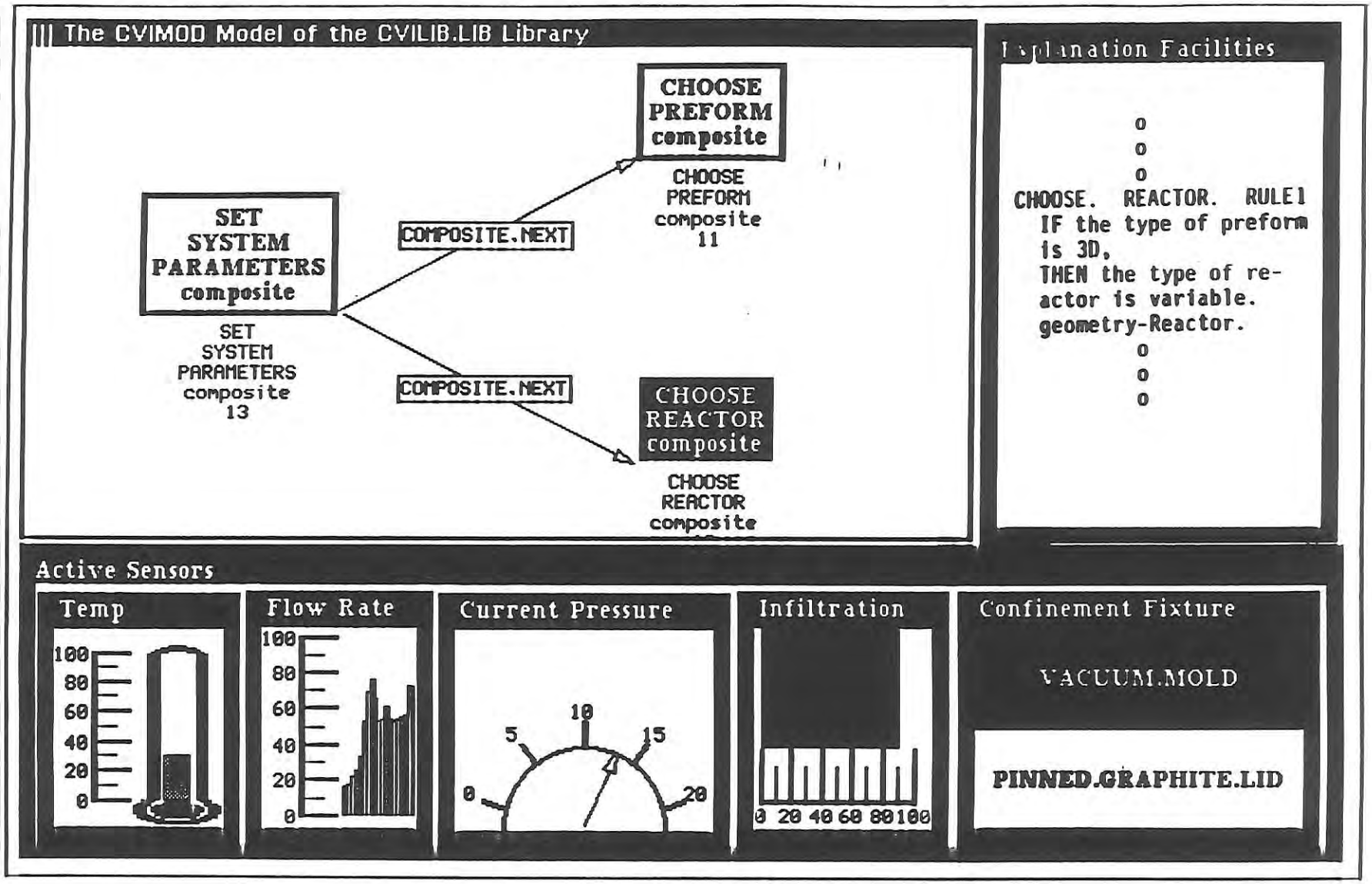


Figure 4.