AI Magazine Volume 6 Number 3 (1985) (© AAAI)

Evolving Systems of Knowledge

N. S. Sridharan

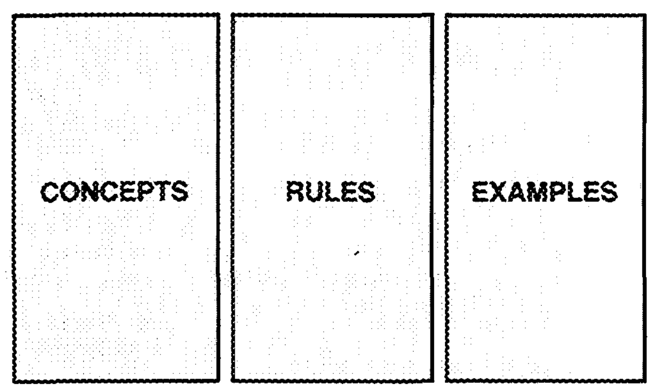
BBN Labs, 10 Moulton Street, Cambridge, Massachusetts 02238

Abstract

The enterprise of developing knowledge-based systems is currently witnessing great growth in popularity. The central unity of many such programs is that they interpret knowledge that is explicitly encoded as *rules* While rule-based programming comes with certain clear pay-offs, further fundamental advances in research are needed to extend the scope of tasks that can be adequately represented in this fashion This article is a statement of personal perspective by a researcher interested in fundamental issues in the symbolic representation and organization of knowledge

We view knowledge based systems in terms of three related spaces: *Concepts, Rules, Examples.* Section 1 deals with knowledge based systems where rules play a prominent role. Section 2 discusses the organization of the rule space. Section 3 displays a wide variety of rule operations in addition to the rule interpreter which applies rules. Section 4 concentrates on the *evolving nature of knowledge* and identifies several forces of change. It includes an argument that systems must be designed to respond well to forces of change.

The final section focuses on how to structure the concept space and the example space to support *evolutionary change.* Concepts need to be specified not only in terms of the more conventional *logical definitions* for well-defined concepts, but also in terms of the *prototypes and deformations* model for amorphous and open-textured concepts. The utility of a well-stocked store of examples and the ability to generate new examples are emphasized. The proper organization of all three spaces and their interrelationship appears to be a good focus for further research in AI.



Knowledge based systems may be usefully viewed in terms of three interrelated spaces.

Figure 1.

Based on the Keynote Speech on Artificial Intelligence delivered to the Canadian Information Processing Society, Session '84, Calgary, Alberta An earlier version of this article was published in the Proceedings of the CIPS/Session '84

This article was written while the author was with Rutgers University Revision of draft performed at BBN Laboratories Research reported here was supported by NSF Grants MCS79-21471, MCS82-03591 and by a grant 5 P41 RR00643 from the NIH I thank John Bresina and Donna Nagel for constructive comments that has resulted in a more understandable presentation Andy Haas, Frank Ritter, Mark Burstein and Albert Boulanger also gave me useful comments for the final version. I am indebted to Professor Ejan Mackaay of the University of Montreal for referring me to the inspiring book by Twining and Miers.

1. Motivation

Expert Systems have demonstrated through a number of successful projects that

- For certain types of knowledge intensive tasks, their reasoning pattern can be characterized as a sequence of decision making steps and such knowledge can be encoded using a uniform format of *rules;*
- When such rules are at an appropriate level of detail, they can be utilized to offer *explanation* for *how* certain hypothesis are arrived at and why certain goals are being pursued;
- The rule form of programming supports *task decomposition* and allows several engineers to work on parts of the task with a greater degree of independence than when using conventional control flow programming;
- To a certain extent, intertwining of task specification and program development is feasible, thus *reducing the time* needed for getting running prototypes.

Figure 1 illustrates types of tasks that typify current expert systems.

Knowledge based systems may be usefully viewed in terms of three inter-related spaces: *Concepts, Rules* and *Examples.* We use the term space because the elements in each are organized systematically by means of relations and transformations that lead from one element to another (intra-space structure). Further, there are interesting relations between the spaces. This framework allows us to understand where basic research in knowledge representation has focused (concepts, rules) and allows us to identify a focus for future research (examples).

We argue that rational design for a knowledge based system must prepare it for change and evolution. To support evolution and change, the space of concepts must be buttressed by the space of examples. We elaborate an argument that concepts need be related to prototypes. Moreover, the ability to generate new examples by deforming known ones is an important capability needed in such systems. This leads to a perspective that calls for a tight coupling between concepts and examples utilizing a theory of prototypes and deformations. This is a useful topic for research.

In the final section, we demonstrate the possible richness of the relations in the example space. Very little is understood presently about these relations in the example space and a fruitful set of research problems remain open.

2. Nature of Rules

The builder of a new expert system for a specialized task has the choice of an array of system building tools coupled with advice on how to go about the process of building one. A good sampling of current tools and their evaluations as well as 'maxims' can be found in Chapters 5 and 6 of *Building Expert Systems* (Hayes-Roth *et al.*, 1983).

The prominence of rules is the striking unity of all these tools. Thus my point of departure for exploring the shape of things to come is an examination of the nature of rules themselves The following definition of a rule, taken from p. 190 of *The Handbook of Artificial Intelligence* is fairly typical.

A production rule is a statement cast in the form "If this *condition* holds, then this *action* is appropriate."

What is the difference between these If-Then rules and the If-Then statements of programming languages? Both are rules *interpreted in context.* Their difference lies in the different ways the context of a statement is structured, modified, and utilized. For programming languages, the interpreter supplies a *control flow* that flows sequentially, takes conditional branches, and goes around in loops. If, at one time point in the execution of the program, several If-Then statements have their condition part satisfied, the interpreter only executes the one statement where control resides at that time. In fact, since only one statement has the control, the condition part of only that statement is evaluated at that time For production rule systems, the context is a *data context.* This context is used to decide which among the enabled rules is to be selected for execution. The data context consists of a working memory which is a symbol structure repeatedly updated by the action of rules. In many systems, the context contains substantial additional information which includes recency of data elements, specificity of data elements, ranking of rules, and other information about the organization of rules as well as concepts used in the task.

On the one hand it is simple to read a rule independently as "If condition holds, then this action is appropriate," but on the other hand it is quite complicated to understand the role of this rule in the overall functioning of the rule system. It is useful at this point to shift the way in which we read a single rule to be "If condition holds, then the computing agent *may* apply this action." The introduction of this modality into the reading of rules prompts us to examine what other modalities might be introduced in such rules.

Forms of Rules

By examining ordinary usage of rule-like statements in human communication, we can readily come up with other modalities to use in rules. Rules can be used to communicate not only Permissions, but also to specify Obligations as well as *Prohibitions.*

A statement of obligation has the reading "If condition holds, the agent *must* do this action." A statement of prohibition has the reading "If condition holds, the agent *must not* do this action." I also envision a hybrid form of rule which is even more useful. It specifies an obligation with choice, "If condition holds, the agent *must* do one of the following set of actions." In order to allow this greater range of rule forms, further research needs to be done on

Type:	Interpretation of measurements
Method:	Hypothesis selection and ranking based on evidence
Characteristics:	Unreliable, incomplete, possibly contradictory data
Examples:	Well-log interpretation, X-ray crystallography
Type:	Diagnosis
Method:	Measurement selection, Interpretation
Characteristics:	Often involves using model of system organization and behavior
Examples:	Circuit faults, Infections and diseases
Type:	Monitoring
Method:	Diagnosis, Corrective Action
Characteristics:	Real-time operation, Reliable functioning
Examples:	Production line monitoring, air traffic control
Type:	Planning
Method:	Composition from subplans and unit actions
Characteristics:	Many complex choices that affect each other
Examples:	Robot planning, route planning, experiment planning
Type:	Design
Method:	Interactive design aids that utilize rules
Characteristics:	Design rules provide constraints as well as guidance; Mapping between structure and function is central; Maintenance of alternative partial designs is quite complex.
Examples:	Circuit design, Architectural design

Table 1.

how to set up a sufficiently strong sense of context so that conflicting obligations and permissions can be sensibly resolved. A good start in this direction can be found in (McCarty, 1983). Further developments in this direction could enable us to specify planning problems coupled with advice on what may be done and what must not be done in any given situation.

Rule forms have been used to model causal processes of organisms as well as mechanisms. Causal rules have been attributed the form "If condition holds then it will cause the following action." Here, not only a subtle change occurs in the modality becoming "cause," but the computing agent is not the agent of the action mentioned in the rule.

This gives rise to yet another form of rules, which allows mentioning the agent of the action specifically. "If condition holds, then *this agent* may/must/must-not do this action." Such rule forms could be used for multi-agent planning (Konolige & Nilsson, 1980).

Some rules are introduced to define useful concepts. If the connective between the antecedent and consequent can be read as "implies" or "if and only if," rules carry logical import. Sets of rules can be organized according to a suitable subsumption or unifiability relation.

Hierarchies of Rule Sets

Large collections of rules accumulate in capturing knowledge in depth for any substantial real world problem. In realistic task domains we cannot make assumptions about the "correctness" and "completeness" of knowledge presented to the program (Sridharan & Bresina, 1982). There is also a need to ensure that increased knowledge is not a liability to the program.

Rules may be parceled according to the identification of what the source of the knowledge is. To the extent that the different sources of knowledge derive from distinct perspectives on the task, their recommendations may in fact not agree. Hierarchical structuring of these sources provides a convenient way of selecting from multiple rules. If the hierarchy is interpreted as a hierarchy of authority, then that (obligatory) rule with highest authority is selected. If the hierarchy is interpreted as one of specialization, then the more specialized source of knowledge will be seen to provide the more accurate, reliable (permissive) rule.

A quick examination of how regulatory statutes are

Function A: Task decomposition and allocation.

A given compound task can be accomplished by solving parts separately and then combining the results. Computing agents must be selected appropriately to match the task that needs to be solved.

Function B: Information sharing or communication.

Rules produce results or hypotheses that are useful to others. Communication is directed or undirected as in blackboard mechanisms.

Function C: Resource Allocation.

Resource allocation is often guided by policy statements that is best separated and associated with specified category of resources, rather be buried in the programs that call for resource allocation and utilization.

Function D: Scheduling, Locking and Other Protocols.

Rules govern proper behavior among agents that can influence each other in the course of their normal behavior.

Function E: Protection and Access Control.

Global control over permission to access and utilize resources and information is effected by these rules.

Function F: Error Handling and Interrupt Processing.

Inevitably rule interpreters produce undesirable or unexpected results. These rules are for recognition and remedying of such situations.

Function G: Descriptive models.

Causal as well as structural rules describe the environment in which the system is operating.

Function of rules in computer software.

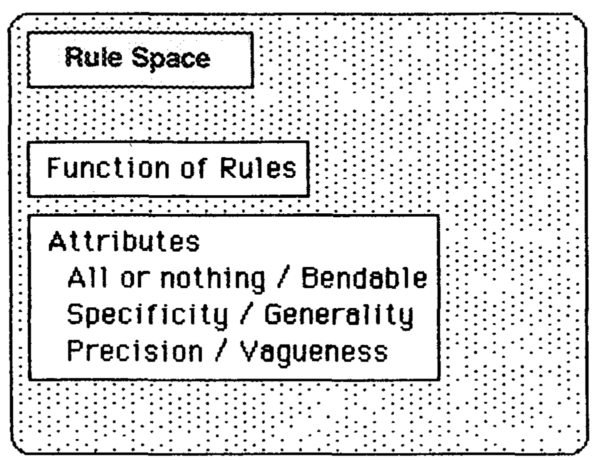
Table 2.

encoded in the legal establishment shows how authority structure and specialization is set up. Rules about the handling of your bank account via electronic banking is governed by federal laws governing interest rates and auditing requirements and income tax disclosure requirements; interstate regulations on banking transactions that cross state boundaries, state regulations concerning the same; the bank policy on hours of operation, frequency of access to account; local branch policy on limits to amounts tendered and overdraft privileges; social conventions about waiting in line for access to the electronic teller; personal preferences about what, where and how much and so on.

Clancey (1983) identifies several other types of hierarchical organization of rule sets aimed at various purposes: consistency at data-directed interpretation, elimination of redundant effort, efficient splitting of hypothesis groups, focusing on problem features, opportunistic triggering, task hierarchy to facilitate planning, context specialization for determining task relevance.

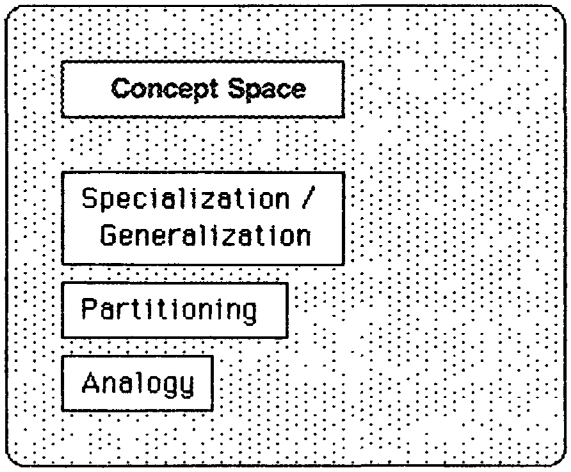
Space of Rules

A set of rules that are structured by relationships among the rules is described as a space of rules. Whereas the discussion of hierarchy emphasizes the systematic relationships that occur among a collection of rule sets, the concept of rule space highlights the internal relationships that occur among the rules within a rule set.



Rule space contains sets of rules, organized according to the function of the rule sets and their attributes.

Figure 2.



Concept space contains a set of concepts organized in familiar ways via hierarchical and associational links.

Figure 3.

Purposes of Rules

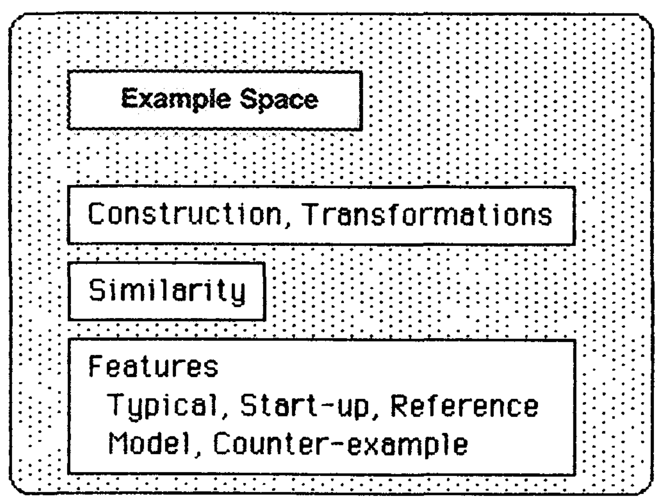
For the present discussion, let us concentrate on the purposes of rules and a few other attributes of rules. Clancey (1983) presents a detailed analysis of the rule sets of MYCIN in an attempt to provide an enhanced capability for explaining. MYCIN's explanations are about how the system arrived at the conclusions and how it plans to use the information being sought. Anyone wishing to understand the system is not likely to find the display of rule applications to be adequate explanation. It is the rules themselves that are in need of explanation. Enhanced explanation capability is achieved by viewing the operation of the rule system as following a strategic plan. Strategic knowledge is embedded implicitly in the internal structure of rules and in the organization of the rule sets. This knowledge is made explicit and cast in domain-independent terms by Clancey. This not only provides economy in representation but also allows explanation to be at an appropriate level of generality.

A central fact that emerges from his study, repeatedly discovered by anyone who has attempted to explain rules, is that rules have purposes which ought to be encoded by the rule author. We present below (Tables 1 and 2) two sets of purposes that arise in two areas of application: Modeling of computer software and legal regulations. The reader may find it interesting to see the parallels in the two sets of purposes.

Attributes of rules

Most researchers who have examined the functioning of organizations, be they corporations, family units, or other institutions like schools or factories, have identified that one of the key elements in understanding the functioning of such organizations is identification of which rules are *bendable* and which rules are *inflexible*. The inflexible rules are all-or-nothing rules that cannot be compromised. For example, in managing a factory job shop, most rules that set policy, involving such criteria as economy, speedy operation, low inventory, allowing time for maintenance procedures and so on, are recognized to be bendable. Each criterion sets up objectives which only provide directions to pursue, but do not dictate absolute objectives to be met. Incorporation of bendable rules into rule systems is investigated by Fox (Fox, 1982, Fox *et al.*, 1982; Fox, 1983).

Multiple rules are often written that cover the same situation, they differ in their *specificity* or *generality*. This can be contrasted with *precision* or *vagueness* of the rules. Specificity refers to the coverage of the rules, that is the situations to which a rule will apply. Precision refers to how well the situations to which a rule will apply can be determined. Rule systems need to be set up so that the internal relations using specificity/generality, precision/vagueness and common/conflicting purposes are used to formulate the *space* of rules.



Example space contains a set of examples, procedures for construction and transformation, similarity measures and a classification of examples.

Figure 4.

Function A: Effecting private arrangements.

Rules about contract facilitate orderly dealing among agents.

Function B: Information sharing regulations.

Free flow of appropriate information is essential to the functioning of a community and these regulations govern this purpose.

Function C: Allocation of public funds.

Regulations are set up to govern how public funds may be used. Regulations are also set up to decide on priorities among communal tasks when resources are limited.

Function D: Regulatory codes.

Various agencies set up as watchdogs prescribe regulatory codes that govern proper behavior by agents. Misbehavior can result in harm that would be hard to rectify; preventive and regulatory rules are used.

Function E: Prohibition and punishment of bad conduct.

This is yet another body of rules governing interpersonal and inter-institutional interaction.

Function F: Grievance and remedial procedures.

The last mentioned, but most visible aspect of law governing courts and their operation. This includes criminal and penal codes.

Function of legal rules and regulations.
Table 3.

3. Rule Handling

So far we have spoken about rules, examining their internal form as well as their attributes and relationships. In the earlier section, we also spoke about the "rule interpreter" emphasizing how it interprets each rule by adding a "context" to it. Whereas rule application is the core of any rule interpreter, there are several other things to do with rules. I draw my inspiration as well as information from a delightful book by Twining and Miers (1982), *How To Do Things With Rules.*

Once a rule set is in existence, a rule interpreter may be used with it. A rule interpreter is typically endowed with the following operations on rules: Rule retrieval, rule interpretation, and rule application. Rule retrieval is the method of selecting, identifying, and testing rules that are candidates for execution. Rule interpretation adds the context information (recall the working memory, data recency, rule ranking information, task or subgoal information) and thereby allows selection of the most appropriate rule to execute. Rule application is the performance of the action specified in the rule. Several alternative methods of organizing all of these operations are recorded in the literature (Forgy, 1979). I invite the reader to ponder several other possible operations on rules (Figure 4) that take place on rules during the life cycle of a rule system.

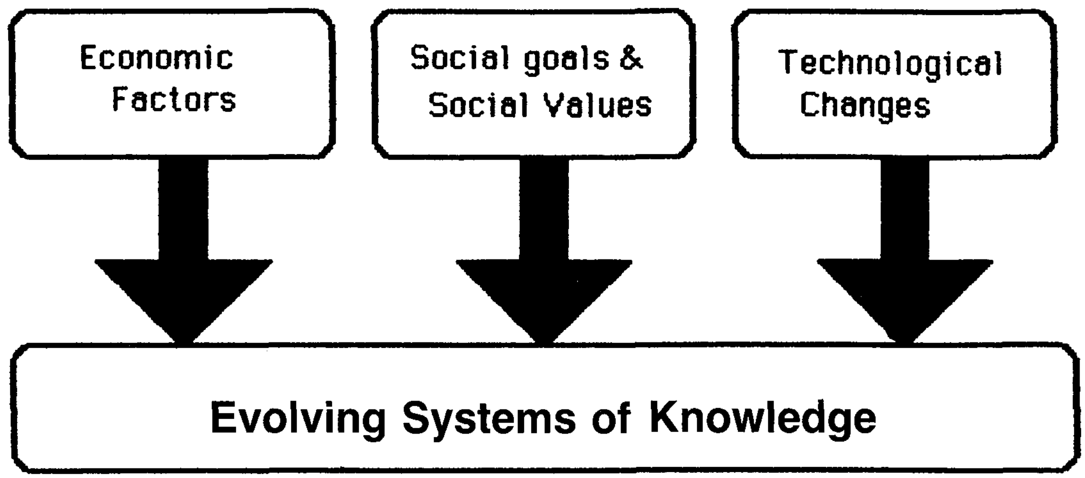
Since we speak of the life-cycle of rules, naturally there are operations that are concerned with the birth of rules, their death, and their change and growth as well as utilization. Adequate explanation of rules requires knowledge of rule attributes such as purpose and specificity. What is the knowledge required to support the rest of the life-cycle activities? Typically, knowledge about the rules and about the rule sets (*e.g.,* the history of rules and their justification, collection of examples that support a rule). Such knowledge often goes under the label *Meta-Knowledge.* In chapter 7 of the book (Hayes-Roth *et al.,* 1983) cited earlier, we find the statement with which we heartily concur:

> The program itself must assume more and more of the burden of understanding its own behavior, documenting and justifying itself, and even modifying itself.... Experts are ... capable of explaining, learning, reorganizing, reformulating, all the while rating the progress they are making.

Another fairly long-term research direction is the construction of facilities to encode such meta-knowledge explicitly. In a later section of this article we address the use of a space of examples to support the rest of life-cycle activities. Let us turn our attention now to the evolving nature of rules during their life-cycles.

Evolving Nature of Knowledge

Any large real-world knowledge base of rules has derived its knowledge from a collection of experts. There is hardly a knowledge base that represents the thinking of one individual. Let us acknowledge that there is great intellectual



Economic, social, technological, and other issues affect how we carry out Interpretation, Diagnosis, Monitoring, Planning and Design tasks in any chosen task domain. Knowledge systems are continually evolving in the short term and in the long term, responding to these forces of change.

Figure 5.

diversity in men's actual modes of thought. The knowledge encoded in a rule system reflects all the diversity of the several experts who contributed to it. For this to reach a stable state each expert's contribution has to be mended, bended, adapted and coerced to reside peacefully with those of the others. Thus there is an *evolutionary process* inherent in the process of the creation, testing and validation of these rules.

Ah! What happens once a rule set is formulated and checked out? I believe that no rational body of knowledge once created and validated to conform to the needs of the moment will last in that condition for long. I agree with the claims of Toulmin (1972) who says (p. 84):

> The rationality of a science (for instance) is embodied not in the theoretical systems current in it at particular times, but in its procedures for discovery and conceptual change through time ...The intellectual content of any rational activity forms neither a single logical system, nor a temporal sequence of such systems. Rather, it is an intellectual enterprise whose 'rationality' lies in the procedures governing its historical development and evolution.

There is an analogy, for instance, in how our emphasis in programming has shifted radically from the "Art of Programming," to the still developing "Art of Software Engineering;" that is, from how to construct programs to how to maintain and modify them. The need for evolution is the very bane of large software systems, which made no provision for this. Current rule-based expert systems capture only a snapshot of the evolving state of knowledge relevant to the task. A system, once built, will be crying out for change and adaptation to keep pace with the continual flux of our social matrix. Thus, the task of expert system builders will not be accomplished by conquering task after task, in the hope of encoding knowledge about all interesting tasks in rule form. Such a journey through the territory of application tasks will not reach conclusion or convergence. The knowledge needed for successful in application areas will continue to evolve long after the snapshots are taken. As Toulmin says, the essential knowledge of a task area lies in its procedures governing change and adaptation and cannot be captured in snapshots.

Rule systems have been advertised to have as their virtue easy *modifiability*. Yet, in practice this is a claim hard to substantiate. True, it seems that rule systems, with their reliance of a data context rather than control context for interpretation of statements, bring us one step closer to this goal of easy modifiability. But we need to understand in greater depth the nature of change and adaptation that will inevitably take place.

Forces of Change

Economic factors exert a great deal of influence on what course of action we select at any time; and these are in constant flux. *Social values* also continually change, sometimes exhibiting discontinuity. Both of these forces require us to continually revise what goals we pursue and how we assign priorities to several competing goals. For example, answers to the following questions reflect a mixture of values and will continually change with time: Whether to use aerosol additives in spray cans, how much to rely on drug-based treatments, whether CRTs give out excessive radiation, to what extent nuclear power plants ought to be sabotage proof, whether auto-emissions ought to be controlled. In addition to these, we find technology itself a source of evolutionary pressure. *Technological changes* are rapidly introducing new instruments with which measurements can be taken and producing new instruments for taking action. When a new instrument is introduced to measure pressure in the human eye, it affects the expert system that was designed to classify occurrences of Glaucoma in patients. When a new technique is designed to reconfigure circuits in wafers after they are fabricated, expert systems for designing circuits need to account for this fairly radical change.

These economic, social, technological and other issues affect how we carry out Interpretation, Diagnosis, Monitoring, Planning and Design subtasks in any chosen task domain. Is the only way to respond to change and adaptation to start afresh and build anew? That would be clearly irrational. The essence of being rational is to prepare for change. We quote Toulmin again:

> A man demonstrates his rationality, not by a commitment to fixed ideas, stereotyped procedures, or immutable concepts, but by the manner in which, and the occasions on which, he changes those ideas, procedures and concepts.

One more specific example is worth noting, just to clarify that I am not speaking only of long-term changes in rather distant social goals or values. I am actually speaking of very real changes that occur within a brief span of time. The span can be so short that by the time the expert system is developed and tested, a change might have actually occurred. The thesis of Karen Kukich (1983) reports on a rule based system, ANA, that was designed to generate natural language reports of the day's stock market activity. It takes as input data from the Dow Jones News Service and produces a three-paragraph summary in English. She cites the following (p.115):

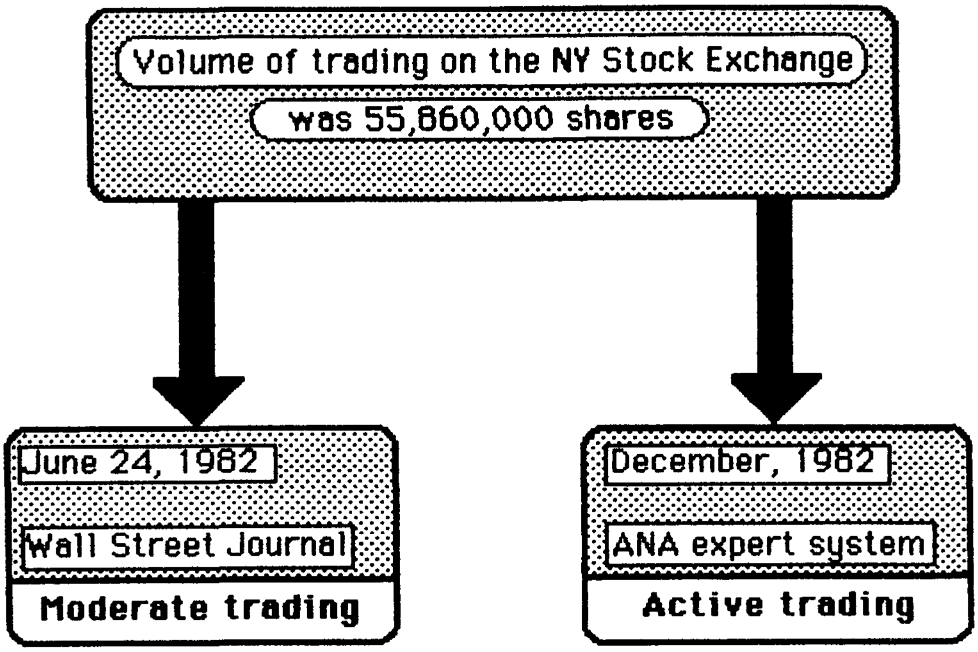
> On June 24, 1982, the volume of trading on the New York Stock Exchange was 55,860,000 shares The Wall Street Journal interpreted this volume as "active trading"; ANA interpreted this as "moderate trading". The reason for this difference is significant. ANA's report was actually generated in December of 1982. In August of 1982, a dramatic surge in the volume of trading on the stock market took place It became a common occurrence for volume of shares traded to break the 100 million level When that pattern persisted, ANA's semantic inferencing production rules were adjusted to reflect the current situation Thus, a trading level of 55 million may have been active by the standards of June 1982, but was more likely to be considered only moderate by December's standards.
>
> This distinction in semantic interpretation over time serves to illustrate the continually adapting and evolving nature of both semantic and linguistic knowledge To be realistic, a knowledge based report generator should be *programmed to adapt automatically* [emphasis added] This would not be difficult in the case of interpreting volume levels It would require a simple calculation to determine the average volume level over the past three months, for instance, each time a semantic interpretation is needed. However, it is not only semantic knowledge that must be adapted to a changing environment, but also linguistic knowledge. Words and phrases go out of vogue, in response to novel developments and novel use of metaphor This remains a more difficult problem to be solved by a knowledge-based report generator.

The force of this example comes from three dimensions. Firstly, it is quite persuasive in displaying a need for system to adapt automatically. Secondly, it suggests language is evolutionary in society—a fact often ignored conveniently in natural language comprehension/generation systems. Finally, it shows that in the short period of time required to engineer a knowledge base (six months in this example) significant changes can already have taken place. Knowledge based systems may ignore evolutionary forces only at their own peril!

5. Concept Space: Prototypes + Deformations

If a perfect rule is one immune from change, there is no perfect rule. What qualities can a rule have that would make it immune from being changed? If a rule can be cast in a fixed form and never changed, we might think of it as being a perfect rule. A rule is perfect if

- it has a single clear purpose;
- it is clearly and precisely expressed and its context of use is clear and precise, leaving no room for doubt or any interpretation;



In the short period of time required to engineer a knowledge base, six months in this example, significant changes can already have taken place. Language is evolutionary in society. Knowledge base systems must adapt to changes.

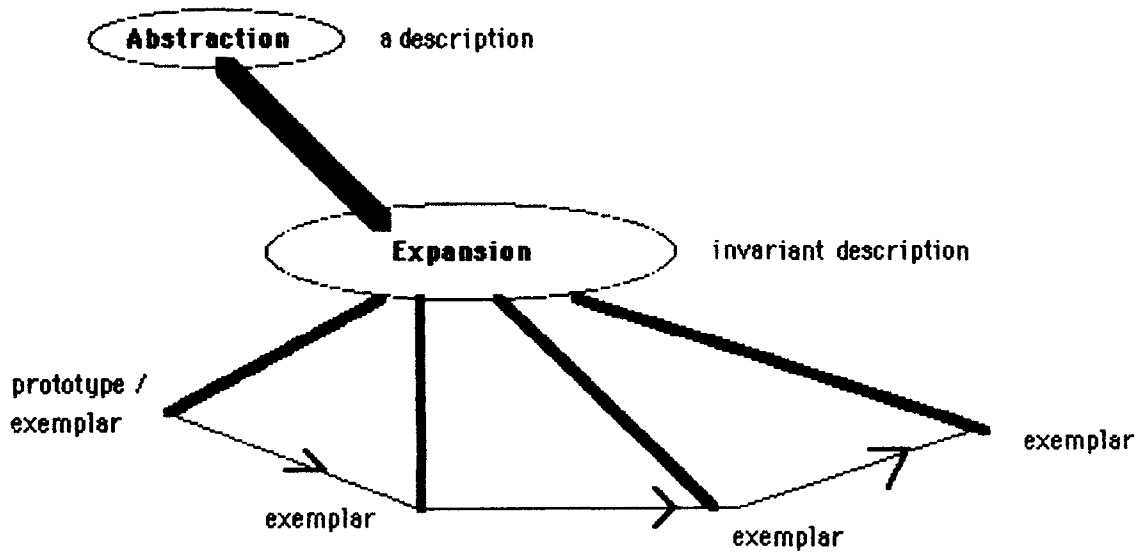
Figure 6.

- it achieves its purpose without causing any (undesirable) side-effects;
- the purpose of the rule is constant. There is no perfect rule; to speak of one is to speak of only idealized fiction.

How might one build a knowledge base to be amenable to change? This question, a central one, may be paraphrased as, How might one build a knowledge representation system in which one could support a wide range of rule-handling techniques (see Figure 4) including: Rule creation, assimilation, interpretation and rule adapting? In this section, we propose that the answer lies in (a) structuring the concept space to have significant connections to the example space, and (b) supporting various forms of example-based reasoning. *Example-based* reasoning includes seeing similarity between sets of examples, judging which similarities and dissimilarities are significant, and generating new examples to meet stated constraints. The remainder of this section advances a proposal that a concept space needs, in addition to the usual logical definitions, a prototypes+deformations representation. There is support for this proposal coming from writers in philosophical logic, philosophy of science, cognitive psychology and philosophy of language. We are adding to these our own arguments based on the idea of evolving systems of knowledge.

Rules and their purposes are formulated using an underlying set of concepts; the meaning and interpretation of the rules depends crucially on the meaning attached to these concepts. Several knowledge representation systems (see especially Tsotsos, 1983) have given attention to the modeling of concepts, especially in terms of the internal structure of concepts (slots, fillers, modifiers, constraints) as well as the relationships that interrelate concepts (inclusion/exclusion, specialization, part/whole, associations, etc.). In defining concepts, however, these systems are predominated by the thesis of "logical definitions," *i.e.*, a concept is defined by an expression that provides "necessary and sufficient" conditions for classifying a given situation as an instance of this concept. In some systems concepts can be defined with weights associated with attributes, and these allow conclusions about concept membership stating a degree of confidence about the conclusion. Yet, the experience we have to date reveals that most concepts in use by us do not have stable necessary and sufficient conditions.

Wittgenstein (1953) argued eloquently about how attributes of instances of "games" are neither individually necessary nor jointly sufficient conditions for recognizing games as such. He used the phrase "family resemblances"



Concepts are "defined" not only in terms of their logical expansion but also in terms of prototypical examples and systematic transformations among them.

Figure 7.

to call attention to the fact that two instances of a concept may share a number of attributes in common thus resembling each other; but all the instances taken together may not have a single attribute in common, thus none of the attributes is a necessary one. Peter Achinstein (1968) has examined concepts in science, typically object and substance names such as "insect" and "copper," measurement terms such "velocity" and "temperature," as well as abstract concepts such as "reversible process" and "rigid body." He concludes that attributes cited in a definition will not be logically necessary nor logically sufficient, but for the most part *relevant*. To reason about the presence or absence of a given attribute and its effect upon the classification of a situation, one must understand the centrality of this attribute for the concept. Lakatos (1976) makes a case that even in mathematics, a term like "polyhedron" is used in conjectures, proofs as well as counterexamples, without it having a fixed or precise meaning. As the meaning changes, the central characteristics begin to be more sharply understood. Our own explorations into the nature of legal argumentation has also helped us rediscover the fact that most legal concepts are notoriously *open-textured* (Hart, 1961). Furthermore, legal rules deliberately introduce hedge words (reasonable length of time, due care, substantially the same condition) diluting whatever might be seen to be central and fixed about terms.

Rosch (1976) has studied the structure of human knowledge about categories of real-world objects, such as guitar, hammer, lamp, eagle. She states that

> Categories are coded in cognition in terms of prototypes of the most characteristic members of the category. That is, many experiments have shown that categories are coded in the mind neither by means of lists of each individual member of the category nor by means of a list of formal criteria necessary and sufficient for category membership but, rather, in terms of a prototypical category member. The most cognitively economical code for a category is, in fact, a concrete image of an average category member.

Benjamin Cohen (1982) has taken such analyses farther, by examining what sorts of computational representations might be useful for capturing such open-texture. He has examined probabilistic as well as fuzzy-set models of the relation between attributes and concepts, favoring more of the structured concept ideas currently being explored in several knowledge representation systems. In our own work (Sridharan, 1982; McCarthy *et al*, 1981), we have undertaken to extend such a structural model, by allowing references to "prototypical" exemplars and knowledge to produce "variations" from them. In deciding whether to classify an instance as a member of a concept, a match is made to a closely matching prototype, but differences and similarities all need to be justified by means of transformations applicable to the prototype. Such a theory attempts to provide a constructive meaning to the notion of fuzziness of instance membership or centrality of attributes. The theory speaks of the "strain" induced by deformations on the prototype and the "strength" of generic as well as specific deformations that enter into the model. The thesis work of Donna Nagel, (in progress) is one attempt to lend computational sense to this model. She also identifies several important algorithms for example-based reasoning.

Space of Examples

Stored examples form the backbone of experiential knowledge that can support rule construction and modification. The utility of a *well organized stock of examples* lies in giving a system the ability to cope with both open-texture and the forces of change. The ability to *generate new examples* from a stock of old ones is also crucial. It is appropriate now to comment on the space of examples and to discuss the structuring relations that run between examples. Rissland (Michener, 1978; Rissland, 1983) has argued for the centrality of this space for all learning systems. In speaking of "constrained example generation" she underscores the importance of generating new examples that suit the purposes that arise.

Rissland illustrates the following types of examples, that are especially relevant to the structuring of mathematical knowledge.

- *Startup examples* are grasped immediately when studying the task domain for the first time; they are highly suggestive of the central ideas and questions to be studied; results can often be lifted from the particular case to the general case reliably. For instance, polynomials of one real variable is a start-up example for the concept of a ring.
- *Reference examples* are repeatedly used throughout a theory and serve to link together many results and concepts. For instance, for the geometry of planar triangles, the set of examples [3-4-5, 30-60-90, 45-45-90, isosceles and equilateral triangles] might serve as reference examples.
- *Model examples* are "canonical" in the sense that they may represent the "general case." In their use, they are almost always adapted and customized to meet the specifics. In plane geometry of triangles, a canonical triangle is often depicted with arbitrary sizes, but with all angles less than a right angle. Gelernter's theorem prover used such examples to advantage, often claiming simple subgoals in the proofs as being true "by diagram." In dealing with conic sections, model examples are drawn with their principal axis aligned with the X and Y axes and the figure centered at the origin.
- *Counter-examples* are often used to delimit the generality of statements. A common use of counter-examples is in showing that the converse of an implication does not hold: for instance, symmetric matrices are diagonalizable. The 2×2 matrix $[\langle 01 \rangle \langle 01 \rangle]$ is diagonalizable but is not symmetric. Thus diagonalizable matrices are not necessarily symmetric.

Producing Variations

A brief excursion into the area of law can be used to illustrate (a) that many concepts are defined primarily through examples; and (b) there are natural ways to produce variations from a given situation. Consider how a corporation is considered not being subject to tax (CCH, 1982).

> Principally there are included non-profit corporations, such as a charitable, religious, or educational institution; a civic league or social club; amateur athletic organizations that do not provide facilities or equipment; organizations forming part of a group legal services plans; political organizations; some cooperatives operating on a patronage basis; an employees' pension or profit-sharing trust; trusts created to satisfy claims for disability or death due to pneumoconiosis under the Black Lung Act; and a trust set up under a plan which provides supplementary unemployment benefits to employees and, as a subordinate part of the plan, may also provide sickness and accidental benefits

This quotation seems to illustrate that the concept of an organization exempt from taxation is specified by *enumerating a list of possibilities.* Different possibilities seem to be specified to different degree of detail. No single necessary or sufficient condition is obvious. If a corporation does not seem to be in exactly one of these, one judges whether it is tax-exempt by looking for similarity to one or more of these cases.

Now let us turn our attention to an illustration of how natural variations can be produced. Take as a specific example the woodstock corporation, a domestic corporation, which has an individual shareholder who owns 100% of the stock of this corporation. He received distributions during the year totalling $20,000 in cash. Variations on this example come easily if we identify several dimensions of variation:

1. Shareholder:
   - The only shareholder is an individual (given)
   - The only shareholder is a corporation
   - The only shareholder is a related corporation.

- There are several shareholders; this shareholder owns majority shares
- There are several shareholders; no one holds majority shares

2. Corporation:
   - Corporation is a going concern and is domestic (given)
   - Corporation is a holding corporation
   - Corporation is not a domestic corporation
3. Distribution:
   - All of the distribution was in cash (given).
   - All of the distribution was in common shares
   - Distribution was mostly shares, plus cash.
   - Distribution was in bonds, shares and cash.
   - Distribution is in shares of a related corporation.
   - Distribution is in shares of an unrelated corporation.
4. Status:
   - Market value of shares has increased or decreased.
   - Distribution exceeds the amount of earnings and profits, so part of the distribution was return of capital and not dividend
   - All of the distribution was return of capital.
5. Other Dimensions:
   - Time over which the shareholder has held shares
   - First ever distribution by corporation
   - Shareholder has redeemed all or part of his original shares.
   - Shareholder has redeemed all or part of his distribution.
   - Distribution was part of a corporate reorganization.

Transformations that can produce such variations are essential when representing concepts by means of prototypical examples. Associated with categories of classification are *annotations* that determine whether a given variation progressively degrades the similarity to the prototype, whether it preserves equivalence, or whether the resulting situation crosses the "fence" thereby becoming an example of a polar-opposite category. For instance, in deciding whether receipt of distribution by this shareholder is "taxable," increase in market value is irrelevant to the classification, and thus preserves equivalence. If the distribution were a mixture of shares and cash, it would be less representative, degrading the prototype. If the distribution were entirely in shares of the corporation, the situation crosses the fence, and the distribution becomes nontaxable.

The organization of the example space is an important research problem. It requires the articulation of attributes of examples (such as those cited previously), a repertoire of transformations and algorithms for using the example space in reasoning.

The extension of the concept space to support example-based reasoning is also a basic research problem for AI. It is my hope that research results along these lines will enable us to produce, in the future, knowledge based systems that would prove viable even in the presence of strong evolutionary forces. Sound human reasoning and good institutionalized decision making are characterized by their ability to anticipate and respond to change We need to explore this dimension for knowledge-based systems.

Summary

Our starting point for this article was to discuss knowledge based systems in which rules occupied a central position (Section 1). We examined in section 2 several forms of rules and commented on the use for diverse modalities of rule statements. We also discussed attributes of rules, such as purpose, precision/vagueness, specificity/generality, bendable rules and so on. We concluded Section 2 highlighting the organization of a space of rules. Section 3 considered a variety of rule-handling operations, some of which need to be elaborated further, if we wish knowledge-based systems to assume greater responsibility for documenting, justifying and modifying itself.

Section 4 concentrated on the evolving nature of knowledge. Economic, social and technological forces of change were identified, some long-range and some short-range. An example (natural language report generation) was used to motivate why systems must be designed to evolve with forces of change, at least for the short-range forces of change. One proposal was made on how to build a knowledge base that is amenable to change, namely, to incorporate example-based reasoning and to structure the concept space to tap into a well-organized example space.

To support example-based reasoning, not only are concepts specified in terms of the more conventional logical definitions for well-defined concepts, but also in terms of the prototypes and deformations model for amorphous or open-textured concepts. The utility of a well-stocked store of examples and the ability to generate new examples was emphasized in section 5. This space of examples has not received much research attention in artificial intelligence. The organization and management of a space of examples is crucial to the construction and modification of rules and more work needs to be done on this.

References

Achinstein, P (1968) *Concepts in science: a philosophical analysis.* Baltimore: The Johns Hopkins Press.

Barr, A & E Feigenbaum (1981) *The handbook of artificial intelligence* Los Altos, CA: William Kaufmann, Inc.

Cohen, B (1982) Understanding natural kinds Ph D thesis, Philosophy Dept , Stanford University

Creation Activities:
Draft, Amend, Make, Adopt, Adapt

Assimilation Activities:
Announce, Promulgate, Communicate, Internalize

Retrieval activities:
Find, Identify, Test applicability, Select

Interpretation activities:
Interpret, View it in context, Apply, Invoke, Distinguish, Rank, Rate

Explanation activities:
State, Expand, Elucidate, Analyze, Explain, Restate, Paraphrase

Attitudes to rules:
Obey, Conform to, Observe, Act on, Test its limits, Disobey, Break, Ignore, Avoid, Evade

Adapting rules:
Twist, Stretch, Manipulate, Restrict, Bend, Waive, Exempt, Enforce, Uphold, Defend, Attack, Criticize, Disapprove, Repeal, Nullify, Abrogate

Things to do with rules!

Table 3.

Clancey, W. (1983) The epistemology of a rule-based expert system. *Artificial Intelligence* 20(3):215-251.

CCH Federal Tax Course (1982) Chicago: Commerce Clearing House, Inc., p. 2014.

Forgy, C L (1979) On the efficient implementation of production systems. Ph. D thesis, Computer Science Dept., Carnegie-Mellon University

Fox, M S (1982) *Job-shop scheduling: an investigation of constraint-directed reasoining* AAAI-82, 155-158.

Fox M S , Allen, B., Smith, S. & G. Strohm (1983) *ISIS: A constraint-directed search approach to job-shop scheduling* Proceedings of the IEEE Computer Society Trends and Applications. National Bureau of Standards, Washington D. C.

Fox, M S (1983) Constraint-directed search: a case study of job-shop scheduling PhD thesis, Carnegie-Mellon University.

Weiss, S M , Kulikowski CA, Amarel S, Safir A. (1978) A model-based consultation system for computer-aided medical decision-making. *Artificial Intelligence* 11(1&2): 145-172.

Hart, H. L. A. (1961) *The concept of law.* Oxford: Oxford University Press.

Hayes-Roth, F. Waterman D, Lenat D. (1983) *Teknowledge series in knowledge engineering.* Building Expert Systems, Volume 1. Reading, MA: Addison-Wesley

Konolige, K & Nilsson, N. J. (1980) *Multiple-agent planning systems* AAAI-80, 138-142

Kukich, K (1983) Knowledge-based report generation: a knowledge engineering approach to natural language report generation. PhD thesis, U of Pittsburgh.

Lakatos, I (1976) *Proofs and refutations* Cambridge: Cambridge University Press.

McCarty, L. and Sridharan, N. S. *The representation of an evolving system of legal concepts II. Prototypes and deformations* IJCAI-7, 246-253.

L. T. McCarty. *A logic of permissions and obligations* IJCAI-83

Michener, E Rissland (1978) The structure of mathematical knowledge PhD thesis, Massachusetts Institute Technology-AI Lab.

Rissland, E L Examples and learning systems Tech. Rep 83-16, University of Massachusetts, Amherst,

Rosch, E (1976) Classification of real-world objects: Origins and representations in cognition. In P N Johnson-Laird and P C Wason (Eds.), *Thinking, readings in cognitive science* Cambridge: Cambridge University Press.

Sridharan, N. S. (1982) *A flexible structure for knowledge* Proceedings of the International Conference on Cybernetics and Society IEEE. Seattle, Washington.

Sridharan, N. S. & Bresina, J L. (1982) *Plan formation in large, realistic domains* Fourth National Conference of CSCSI/SCEIO, 12-18. Canadian Society for Computational Studies of Intelligence Societe Canadienne pour Etudes d'intelligence par Ordinateur

S. Toulmin (1972) Human understanding: the collective use and evolution of concepts. Princeton, NJ: Princeton University Press.

Tsotsos, J K (1983) Knowledge organization: its role in representation, decision-making and explanation schemes for expert systems Tech. Rep. LCM-TR83-3, University of Toronto, Dept of Computer Science,

Twining, W & Miers, D. (1982) *How to do things with rules: A primer of interpretation, Second Edition.* London: Weidenfeld and Nicolson Distributed by Fred B Rothman, Littleton, Colorado.

Wittgenstein, L. (1953) *Philosophical investigations* New York: Macmillan.