NOT FOR QUOTATION
WITHOUT PERMISSION
OF THE AUTHOR

EPISTOMOLOGICAL ASPECTS OF KNOWLEDGE-BASED DECISION SUPPORT SYSTEMS

Ronald M. Lee

July 1982
WP-82-78

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

ABSTRACT

Knowledge-based decision support applications differ from those typical of artificial intelligence expert systems in their open-ended, evolutionary character and need to coordinate with other systems resources, such as organizational databases and quantitative analysis routines. While knowledge representation machinery is becoming available, the corresponding formalization of managerial/administrative knowledge needed for DSS application is still lacking.

This entails problems of an epistomological nature, identifying the foundational concepts of business. An abstract framework based on formal languages and denotational semantics is proposed, and ontological issues are identified.

Keywords: decision support systems, knowledge representation, knowledge-based systems, applied epistomology, denotational semantics

CONTENTS

EPISTOMOLOGICAL ASPECTS OF KNOWLEDGE-BASED DECISION SUPPORT SYSTEMS

Ronald M. Lee

I. INTRODUCTION

The influence of artificial intelligence (AI) in decision support systems research has now become an identifiable trend. This draws mainly from AI work in knowledge representation and expert systems. The book by Bonczek, Holsapple and Whinston (1981) provides a good background reference.

The question arises as to the difference between a 'knowledge-based' DSS which uses AI type knowledge representations and an AI expert system using similar mechanisms. The basic distinction is in the system's objectives. An expert system seeks to replicate, hence replace the abilities of a human expert in specific problem domain. A knowledge-based DSS on the other hand seeks to assist a human (manager) by taking over the more structured parts of a larger, only partially formalizable, problem domain.

It is here that the basic concerns of this paper arise. Expert systems typically involve a closed-world assumption; the problem domain is circumscribed, and the system's performance is confined within those boundaries. In DSS contexts, on the other hand, the world is open. A knowledge-based DSS must be adaptable and extendable to meet the evolving needs of the user and changing conditions in the environment.

The author wishes to acknowledge useful interactions with Helder Coelho, Steven Kimbrough, and Amilcar Sarnadas on these topics.

More importantly, it is clear that DSS's oriented towards individual users are only a special case of the much broader problem of aiding *organizational* decision processes. This raises the important problem of interactions between knowledge representations, an aspect largely ignored in AI.

Why is this a problem? The reason is that knowledge representation schemes, e.g., semantic net formalisms, various forms of predicate calculus, have been designed to be *general purpose*, be applicable in any variety of subject areas. Thus, each new effort at knowledge base construction must essentially start from scratch, and the semantic elements chosen tend to be ad hoc, specific to the immediate problem at hand. Consequently, efforts to extend or modify the knowledge-base for changes in the problem scope or definition, and attempts to interface the knowledge-base to other knowledge-bases, databases, etc. are usually frustrated by semantic incompatibilities. (Similar criticisms apply to the design of databases, leading to the semantic difficulties of database translation.)

On the other hand, the contention here is that managerial applications do have certain commonalities (they must or business schools would have nothing to teach), and that these commonalities, properly formalized, can guide and discipline the design of knowledge bases in managerial domains. The issue becomes one of *epistomology* – seeking the basic semantic foundations upon which managerial knowledge can be constructed.

In the sections to follow, the potential role and character of a knowledge-base in a DSS is discussed. The use of knowledge bases in DSS applications poses two types of problems not typically in artificial intelligence contexts: one, a broader, open-ended and evolving problem domain; and two, interactions with other system resources (databases, quantitative routines). In order to focus on the theoretical issues involved, an abstracted view of a DSS as a formal language is proposed. This highlights the fundamental role of a uniform semantic foundation (ontology) for the various DSS components. Using this perspective, various issues in philosophical semantics are described as they apply to managerial DSS applications.

II. STRUCTURE OF A KNOWLEDGE-BASED DSS

Sprague (1980) characterizes a DSS as having two basic types of problem oriented resources:

1. databases – which contain *facts* about the environment
2. models – which enable *inferences* to be made.

Practically speaking, the models are almost always quantitative algorithms, typically providing optimization or statistical inferences.

A *knowledge- based* DSS adds an additional component, the so-called 'knowledge-base.' The formalisms employed fall roughly into two general categories: semantic net and predicate logic formalisms. The pros and cons of each are much debated, the general objectives are similar: the *declarative* representation of (mainly) *qualitative* knowledge.

Databases, of course, contain both qualitative and quantitative data. Where as operations research models may provide inferences on the quantitative data, a knowledge base provides structures of inference of a qualitative sort.

The more important aspect is that these are *declarative*, as opposed to *procedural*, structures. That is, the problem-oriented information is represented as independent, axiomatic rules which are searched heuristically. While this is computationally less efficient, it is correspondingly more flexible in that a complex network of potential inference paths is represented. It is this aspect which warrants the comparison to human knowledge, capable of being applied in various directions and forms, rather than limited to a single deductive path as are normal (procedural) computer programs.

However, declarative representations are computationally practical only for a limited number of primitive qualities.

Quantities, regarded as qualities mapped onto a linear (ordinal, interval, ratio) scale represent large families of qualities. Thus, represented as declarative axioms, arithmetic becomes terribly cumbersome computationally. This is why such declarative languages as PROLOG have so much difficulty incorporating arithmetic operations and, correspondingly, why quantitative inference is nearly always represented procedurally.

The potential of a knowledge base in a DSS is to provide a unifying framework of higher level abstractions of the qualitative facts in databases as well as incorporating the specialized inferences of quantitative models where appropriate. The knowledge base would thus provide an conceptual map of the user's problem domain allowing flexible and adaptive integration of system resources.

On the other hand, while artificial intelligence research is providing the mechanisms for building knowledge bases, the successful application of these tools depends on a formal understanding of managerial problem domains. This is so far lacking. The need is for an applied epistomology of the knowledge typical in business environments.

III. DSS AS A FORMAL LANGUAGE

The issue here for DSS, as we see it, is to find a representational perspective that somehow avoids computational preoccupations and focuses on the conceptual organization of the DSS in modeling managerial problem domains.

A useful approach is that used in logic for comparing and evaluating logical representations (e.g., van Fraasen 1971). This is to regard each as an instance of a *formal language*, consisting of:

a. *syntax* comprising
   i. a *vocabulary* of elementary symbols
   ii. *formation rules* which define well formed expressions in the language.

b. *transformation rules* — which define truth preserving substitutions between expressions

c. *semantics* indicating what the symbols and expressions of the language denote.

Thus, various logics are compared based on differences in their syntax, inferential power (transformations), and semantics. A similar concept of formal languages is also familiar in theoretical computer science. Turing's concept of abstract automata is as a recognizer of formal languages of varying degrees of syntactic complexity. This view is almost entirely syntactic however. (See, e.g., Hopcroft and Ullman 1974).

When semantics is discussed with respect to computer languages, what is usually intended is *computational semantics:* the machine operations and data structures corresponding to each high level expression. (In human terms, this would be analogous to the neurophysiological representation of our spoken sentences.)

Logicians and linguists are on the other hand concerned with *denotational semantics*, the objects or sets of objects which symbolic expressions signify in the real world. It is this latter concept of semantics which is of concern here.

Earlier we categorized the internal resources of a knowledge-based DSS as:

- databases of quantitative and qualitative facts
- procedural routines for quantitative inference
- declarative structures for qualitative inferencing

In principle, these various components should each contribute to aiding the user's understanding of a certain problem domain. But how do these components interact? A way of examining the problem abstractly is to regard them as various interacting formal languages, or indeed as different aspects of a single formal language.

Clearly the syntactic compatibility of these aspects will be important, though this is mainly an engineering problem. The deeper problems are semantic: how the symbolic expressions of the various DSS components refer to the phenomena in the user problem domain.

IV. MODELS OF FORMAL LANGUAGES

While we normally consider the semantics of a language to be something fixed, it is clear that the association of an arbitrary symbol to the object it signifies is a matter of convention ("a rose by any other name would smell as sweet"). In the perspective of formal languages, this convention is made explicit in the concept of a *model*, which is an assignment of interpretations to the basic symbols of the language. Note that this use of the term 'model' is slightly different than the colloquial usage in the DSS literature. Most of what are there called 'models' would here be called an algorithm that is, they are procedures for performing a sequence of deductions. For instance, a multiple regression routine, in itself, would be an algorithm. However, when an interpretation is given to its terms, e.g., as sales, advertising costs, disposable income, it is then a model in the formal language sense; i.e., it models or is an abstraction from some real world situation.

This usage also differs from that in database management, e.g., the relational or network models. In the formal language sense these would only be models when used to describe some actual organizational environment.

Despite the confusion it may create in terminology, we believe that this formal sense of the term 'model' represents a central issue for DSS research: *that is, to develop a theory which defines families of models (interpretations of formal languages) common to administrative contexts and their variations in specific situations.*

The contributing disciplines of DSS — e.g., database management, operations research, statistics, artificial intelligence, logic, etc. — can be viewed as offering various types of uninterpreted formal languages. These are normally interpreted in specific, isolated situations, for instance, a database design for a bank, an OR model of traffic flows, a regression forecast of sales in a particular market area. *Modeling* (the interpretation of these formal languages) is not itself formalized in these disciplines and remains the art of the technical analyst.

The contention here is that while the phenomena of managerial environments varies widely from one situation to another, there are nonetheless commonalities which can be organized to guide and discipline the modeling process. This organization would no doubt take the form of similarity hierarchies where situations are compared at varying levels of abstraction. Strong evidence for this possibility is the long success of the practice of accounting in providing abstract measures of business activity; e.g., the comparability of financial statements. Accounting however is mainly concerned with measurement, based on monetary valuation, and leaves the underlying phenomena to be informally understood (for instance, few accountants can give a formal definition of an "asset") whereas it is these latter aspects that are the focus here.

V. ONTOLOGY

Ontology refers to the nature of the primitive entities which the expressions of a (formal) language denote; i.e., what basic conceptual constructs are used to define the sets of the objects which form a model of the language.

The purpose of an ontology is to *clarify*, through reduction of informal description to a smaller set of more sharply defined terms. The inferences made in the language can only be as sound as the underlying ontology. (This is a philosophical version of 'Garbage-In-Garbage-Out).

An ontology can only clarify if the sets it comprises (the denotations of the language) are clearly understood and whose elements are clearly distinguishable by the users of the formal language. Thus the adequacy of an ontology is a matter of *consensus*; but it is a consensus that must be carefully scrutinized, since the value of further definitions and inferences in the language depends on the soundness of this foundation.

Since sets consist of discrete individual elements, the central issue in most ontological debates is the identification of individuals. That is, what are the sorts of things (individuals) which form the sets our concepts refer to? An intuitive test for the consensual recognition of individuals is whether the parties involved agree that two individuals are the *same*.

Discrete physical objects, for instance, seldom give rise to confusion, and it is noteworthy that most operations research models apply to ontologies of this type; e.g., involving employees, machines, or physical inventories.

Transformations on physical individuals can however give rise to potential confusions (which gives some insights to the difficulties in dynamic modeling). A delightful example (Brachman, personal conversation) is that of a wooden boat and we replace one of its planks with a new one. Is the modified boat the same individual as the original? Most people would agree. Suppose we continued to systematically replace planks in the boat with new planks until all parts of the boat were now replaced. Is this individual the same as the original? Some, though perhaps not all would agree. Now, suppose we collected the planks we removed and constructed another boat in the design of the original boat. Is it now the same as the original?

Austin (1970) summarized the matter by observing that *similarity* is a property of nature whereas *sameness* is a matter of linguistic usage. The boundaries of individuation, in short, depend on the consensus of the user group or population.

Time spans —e.g., days, weeks, months, years —tend also to be relatively unproblematic in ordinary situations. Few people disagree about the temporal boundaries of 7 December, 1941, for example, despite the minor problems created by different time zones. (Among theoretical physicists, however, the ontology of time is quite different and more open to dispute.)

The ordinary language use of "same" has another, apparently separate sense. "I drive the same car as John," may mean that there is one individual vehicle that we share, or that we drive the same type of car. These are sometimes distinguished as sameness of individuals vs sameness of type. In a logical notation, the latter involves a predicate variable, i.e.,

$$\exists X \; drive(me,X) \;\&\; drive(john,X)$$

vs

$$\exists TYPE \; \exists X \; \exists Y \quad drive(me,X) \;\&\; drive(john,Y) \;\&\; TYPE(X) \;\&\; TYPE(Y).$$

(Here and throughout, constants are lower case, variables upper case.)

As we move out of the domain of discrete physical objects, individuation becomes less clear. For example, in a hospital if a doctor declares that patient Smith has the same disease as patient Jones, it is apparently meant that the two diseases are of the same type, e.g., that the bacteria are of the same species. On the other hand, it may mean that the two diseases are from the same bacterial pool. The difference matters where contagion is of concern. Again it depends on the needs of the user group.

Abstract objects are notoriously difficult to individuate, essentially because there are no lowest level 'atoms' (molecules, cells, etc.) to which one can take recourse. For instance, to say that X independently had the same idea as Y, or that X plagarized or stole Y's idea is extremely difficult to pin down; is this sameness of individuals or sameness of type?

Strawson (1959) asserts that the only reliable basis for individuation is to locate the individual in a spatial temporal framework. In this way, ideas might be identified to the mental activities of a certain person throughout a certain period in time.

These aspects of individuation are of central importance to the development of knowledge-based DSS since, in most cases, these have ambitions to include expertise beyond the ontologically safe domains of discrete physical objects.

VI. ONTOLOGY FOR QUANTITATIVE MODELS

Pure mathematics usually adapts some abstract set of numbers in their ontology, e.g., the integers, real numbers, rational numbers, etc. Applied mathematics, on the other hand, usually includes a broader ontology, namely that the numbers involved are *measures of some scalable properties*. The type of scale involved, e.g., ordinal, interval, ratio, determines the algebraic flexibility of the inferencing. Typically left implicit or informally described, are the individual objects to which these measures are applied. As observed earlier, these are typically straightforward from an ontological standpoint, so little confusion arises.

However, when measures are applied to less obvious phenomena, the summary statistics generated from these measures can become quite ambiguous to the people using them. This has become a serious problem in accounting where monetary valuations are applied to a wide range of disparate phenomena (with subsequent allocations, prorations, amortizations, price level adjustments, etc. applied to them) so that the final results are only vaguely meaningful. For example, an occasional student exercise in financial accounting is to revise a company's net income 100% entirely through adjustments conforming to Generally Accepted Accounting Principles. Knowledge-Based DSS's applied to such domains would be prone to similar difficulties. The suggestion is to expand the ontology to explicitly recognize the types of underlying entities being measured.

VII. ONTOLOGY FOR DATABASES

In the architecture for a knowledge-based DSS presented earlier, current *facts* about the environment are recorded in (one or more) databases. Since these provide the basis for higher level inferences, the ontology they assume plays a fundamental role.

Codd's (1970) Relational Data Model ('model' in the database sense) is often regarded as a useful, mathematically abstracted prototype of database systems. The relations involved are tuples of elements drawn from sets of *data* items (in relational terminology called domains), such as single characters, character strings, integer numbers, floating point numbers, etc. The operation of these systems depends only on the symbolic shape of these items, not on their significance to the users of the system. This is similar to the use/mention distinction in natural language semantics. E.g., the teacher's question

Can you spell "can"?

first uses the word "can," then mentions it (as was done again in this sentence). Database designs present a syntax of data but no denotational semantics. Hence, databases have no explicit real world ontology. However, they often, implicitly, reflect a certain ontology in the definition of relations. For instance, a database

EMPLOYEE(E-NAME,EMP-ID,AGE,...)
DEPARTMENT(D-NAME,DEPT-ID,LOCATION,...)
WORKS-FOR(EMP-ID,DEPT-ID)

implicitly recognizes employees and departments as individuals, with "WORKS-FOR" as a two place predicate relating them. The existential implication is that for each tuple in the EMPLOYEE relation there is an actual employee in the company, and for each tuple in the DEPARTMENT relation there is a department in the company. Such existential presuppositions of certain database relations are the basis of Chen's Entity-Relationship Model (1976).

VIII. GRANULAR AND LIQUID OBJECTS, MASS OBJECTS AND PROBLEMS OF INDIVIDUALIZATION

The world, according to Quine (1960), consists of middle size objects. Problems of individuation arise when we consider granular objects, such as corn, wheat and liquid objects, e.g., water, oil, etc. The problem is the same in both cases: to discretize these objects and assign names to them, it is impractical to go to their lowest level elements (grains or molecules).

While this poses a difficult theoretical problem (logics over continuous domains, paradoxes arising from axioms of choice), in commercial practice, the problem is typically avoided through the simple device of a *container*. That is, these substances are normally conveyed in a (middle sized) container which is easily individuated and named. The contents of the container become properties of (predicates applied to) the container. Emptying one container into another involves changes of properties of the two containers (see temporal aspects, below).

Note that whether something is to be treated as a granular substance or as discretely identifiable objects depends on the interests of the potential users of the language. For instance, rock and gravel companies would no doubt regard stones beneath a certain diameter as granular. A rock collector, on the other hand, would regard them as individually identifiable specimens.

Mass objects are an intermediate class sharing properties of discrete individuals and liquid objects. Examples are planks of lumber, bars of steel, etc. These can be divided into increasingly smaller units of the same substance. These can of course be treated as individual objects. Divisions of the object cause the destruction of the original and the creation of two new individuals. Alternatively, these are often regarded in a way similar to liquid objects, where the container is some specified inventory location, section of a warehouse, etc. In this case, e.g., lumber is treated as so many board feet without regard for how many individual pieces the inventory contains. The choice, again, depends on the intended usage of the formal language.

IX. AN ONTOLOGY INCLUDING TIME

Time, which is so central in commercial environments has, oddly enough, had relatively little development in the concept of formal, especially logical, languages. Principle works on temporal logic are by Prior (1967) and Rescher and Urguhart (1971).

The implicit conception of time in commercial environments seems to be a continuous dimension of time points. This would normally cause the same logical problems as liquid objects except that the reference to time is inevitably with reference to *time spans*, which have a similar ontological status as containers to liquids.

Examples of individual time spans are:

The year: 1984
The month: January, 1981
The day: 7 December, 1941
The minute: 11:59 a.m., 2 July, 1982, Central European Time

An ontology of time might alternatively assume a time line of discrete units of some minimal size. Such is the perspective in digital watches and computer clocks. Time, taken as discrete or continuous, is regarded as linearly ordered. This is the basis for concepts of change and of precedence in changes. By including time in the ontology, the truth of a predicate becomes dependent on time. This amounts to adding a temporal sort to the language and adding a time place to each predicate.

X. POSSIBLE WORLDS SEMANTICS

No doubt the most seductive yet controversial concept introduced in ontological theories this century is that of a *possible world*. Intuitively speaking, a possible world is like a formalized gedanken experiment: it is an imaginary locus to which truth values can be attached. The world we know is a priviledged possible world: the actual world.

Debate over possible worlds centers on whether the concept can be consensually understood sufficiently well by the users of a formal language whose semantics depend on it. (In this regard it is like the utile in economics: theoretically very useful but ontologically rather questionable).

A principal motivation for the concept of possible world is to give a denotational semantics to generic concepts. We would like to consider the denotation of a predicate as the set of things of which it is true. However, those things existing in the actual world are typically not enough. This denotation in many cases must be extended to possible worlds as well.

For example, consider the denotation of the concept: person?* Is the property of personhood equivalent to elementhood in the set of all people currently alive? or the set of all people who have ever lived? or the set of all people who ever lived or will live? Normally, even this last set is considered incomplete, for it refers only to actually existing persons in the past or future. The essence of the concept person (called its *intension)* is however the denotation (or *extension)* of human individuals in all times in all possible worlds.

Further, of perhaps more practical consequence, the concept of possible worlds permits the formal definition of concepts of *action* and *responsibility*. Various conceptions of action are possible, depending on the purpose of the formalization. One, due to von Wright, distinguishes action from a simple change in state in that it is brought about by some (human, organizational) agent. This contains an implicit counter-factual: that if it were not for the agent's intercession, the change would not have taken place. Thus, while a concept of change can be described in terms of transitions in states of the actual world from one time to the next, a concept of action requires the notion of another, possible world to express the state of affairs were it not for the agent's intercession. Thus, by asserting someone responsible for a particular state of affairs, we allude to some alternative state that would exist had that person's influence not been present.

XI. PREDICTIONS, PLANS AND PROMISES

In discourse relating to administration, finance and commerce, it is only statements concerning the past and present that are considered factual. For instance, that company X sold company Y a piece of equipment Z on date D, is either true or false if D is in the past. However, if D is a date in the future, the statement is not regarded as either true or false, but rather one of conjecture or speculation.

Three principal types of conjectures or attitudes in these contexts are predictions, plans and promises. In their semantic formulation, each of these makes an assertion about some possible world in the future, with the additional claim that the actual world will eventually match this possible world.

A *prediction* is simply a description of such a future possible world with the assertion that the course of events in the actual world will eventually lead to this state.

* technically: $\mathrm{den}(\lambda X\ \mathrm{person}(X)) = ?$

A *plan* is a prediction augmented with intentions of action. The assertion is that the future possible world described in the plan would not normally come about, except for the intended actions of the planner.

A *promise* is a plan augmented with a *commitment* to another party. Implicit in the notion of commitment is some penalty for not carrying out the plan. This penalty may be a vague moral reproach, some type of legal recourse or perhaps definite consequences such as foreclosure or siezure of assets.

A promise is the act of incurring an *obligation*. Obligation is one of several operators in a so-called *deontic logic* (von Wright 1968). Others are permission and prohibition. Each involves two parties and an action. Symbolically,

obliged(X,Y,A)	=	X is obliged to Y to do A.
permits(X,Y,A)	=	X permits Y to do A.
prohibits(X,Y,A)	=	X prohibits Y to do A.

These are inter-definable: to be permitted to do something is to not be prohibited from doing it and vice versa; to be obliged to do something is to not be permitted not to do it and vice versa.

A *contract* is a relationship of mutual obligation. A *contingent obligation* is one where the obligation depends on the occurrence of some event. A familiar example is insurance.

Interestingly, the deontic relationships of obligation, permission and prohibition are, in commercial contexts, often reified to the status of *objects*. Examples of deontic objects based on obligation, are notes, various types of bonds, and with a real but less definitely described obligation, the various types of preferred and common stock. Insurance policies are examples of contingent obligations. Examples of deontic objects based on permission are licenses, easements, etc. whereas examples of deontic objects based on prohibitions are: copyrights and patents.

A formal device to accomplish this reification to objecthood is the intension operator, $^{\wedge}$, due to Montague (best explained in Dowty (1981)). (This is essentially a lambda abstraction on time/possible world pairs, serving to make intensions extensional. In our case this operator would be applied to deontic expressions.)

These deontic objects constitute *assets*, that is they are *owned*, by one of the parties involved. For instance a bank owns its notes outstanding; investors own their stocks and bonds. Likewise insurance policies, licenses, copyrights and patents are owned. In the case of promissory objects (deontic objects based on obligation), the object represents a *claim on assets* to the other party.

XII. CONCLUDING REMARKS

In the foregoing we have argued that an important theoretical problem for knowledge-based DSS in organizations involves the epistomology of management: identifying the foundational concepts of managerial knowledge. In this paper we sketched an approach using denotational semantics, and suggested several basic types of individuals: physical objects, numbers, time and possible worlds. We stressed that these basic entities are not to be considered as 'essential' in that no other bases are possible. Rather as Goodman (1978) points out in *Ways of Worldmaking*, all such conceptual systems are a matter of consensus and utility to its user population. However, this does not mean that no generally useful conceptual foundations are possible for managerial domains. Indeed, the widely accepted terminology of accounting provides informal evidence that this is possible. For more detailed discussion of these issues, see Lee (1981a).

REFERENCES

Austin, J.L. 1970. *Philosophical Papers*. Second Edition. (Section 5. Truth, pp.117-133.) Oxford: Oxford University Press.

Bonczek, R.H., C.W. Holsapple and A.B. Whinston. 1981. *Foundations of Decision Support Systems*. New York: Academic Press.

Chen, P.P. 1976. The Entity-Relationship Model — Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1(March):9-36.

Codd, E.F. 1970. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13(June):377-387.

Dowty, D.R., R.E. Wall and S. Peters. 1981. *Introduction to Montague Semantics*. Boston: D. Reidel Publishing Company.

Goodman, N. 1978. *Ways of Worldmaking*. Indianapolis, IN: Hackett Publishing Company.

Ijiri, Y. 1967. *The Foundation of Accounting Measurement*. Englewood Cliffs, NJ: Prentice-Hall. Reprint ed., Houston: Accounting Classics Series, Scholars Book Co., 1978.

Hopcroft, J.E. and J.D. Ullman. 1969. *Formal Languages and their Relation to Automata*. Reading, Mass: Addison Wesley.

Lee, R.M. 1981a. CANDID Description of Commercial and Financial Concepts: a Formal Semantics Approach to Knowledge Representation. WP-81-162. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Lee, R.M. 1981b. Relational Databases, Logical Databases and the Entity-Relationships Approach. WP-81-164. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Prior, A. 1967. *Past Present and Future*. Oxford: Oxford Press.

Quine, W.V.O. 1960. *Word and Object*. Chambridge, MA: MIT Press.

Rescher, N. and A. Urguhart. 1971. *Temporal Logic*. Vienna: Springer-Verlag.

Sprague, R.Jr. and G. Fick. 1980. *Decision Support Systems: Issues and Challenges*. Oxford: Pergamon Press.

Strawson, P.F. 1959. *Individuals*. Methuen & Co. Ltd. Reprint, Anchor Books, Doubleday & Company, Inc., New York in 1963.

van Fraasen, B.C. 1971. *Formal Semantics and Logic*. New York: The Macmillan Company.

von Wright, G.H. 1968. An Essay in Deontic Logic and the General Theory of Action. *Acta Philosophica Fennica* Fasc. XVIII. Helsinki. pp. 293-301.