International Institute for
Applied Systems Analysis
IIASA www.iiasa.ac.at

Data and Language in Organizations: Epistemological Aspects of Management Support Systems

Lee, R.M.

IIASA Working Paper

WP-83-046

April 1983

✓

DATA AND LANGUAGE IN ORGANIZATIONS: EPISTEMOLOGICAL ASPECTS OF MANAGEMENT SUPPORT SYSTEMS

Ronald M. Lee

April 1983
WP-83-46

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

DATA AND LANGUAGE IN ORGANIZATIONS:

EPISTEMOLOGICAL ASPECTS OF MANAGEMENT SUPPORT SYSTEMS

Ronald M. Lee

CONTENTS:

ABSTRACT

This book contributes to the literature on management decision support systems (DSS). DSS research is motivated by the observation that much of what managers do involves unstructured problem solving. For this reason, the structured, procedural models implemented in management information systems (MIS) have had little impact on actual managerial practice.

Actually, the terms 'decision' and 'problem solving' over-simplify the image of managerial activity, if what is meant is choosing from a set of well-defined alternatives. Management also includes such aspects as reality testing, problem finding, scenario generation, and just plain muddling through. A broader conception of management *cognition* – of which decision making is only a part – is therefore adopted. The challenge to technology development is to *support* these unstructured managerial activities. The emphasis is to amplify managerial cognition and to improve decision effectiveness. However, to achieve this we must go beyond platitudes and come to a better understanding of what managers actually do.

The activity of managers is almost entirely linguistic. Computers, as symbolic processors, ought to be an effective complement. However, a fundamental problem, stressed repeatedly throughout the book, is semantic change. The context of managers is always changing, whereas computational inference depends on fixed semantics. Herein lies the basis for a theory of management support systems. The theory takes the form of an applied epistemology: how do managers know their world and detect its changes?

Thus, while this book is oriented towards improving information technology, its attention is primarily to the *content* of management information and only secondarily to technology. Technological innovations abound. What is needed now is a better understanding of what these technologies are to do.

CHAPTER 1:

MANAGEMENT DATA AND LANGUAGE

CONTENTS

> A practical theory for Decision Support will not emerge until we are willing to deal with more profound conceptions of human decision making. Each of us should take care to become aware of the cultural limitations inherent within a "technical" or "engineering" orientation. Each of us should recognize, with deep humility, that our fundamental values may tragically disable our honest mission of improving managerial decision making.*

A. INTRODUCTION

The use of information technology to aid management has been of interest since the early days of computing. Yet, contrary to expectations, the activities of managers have been relatively unaffected, notwithstanding enormous advances in the technology. Managers, for their part, are often keenly disappointed at this. The velocity and complexity of managers' activities seem to keep increasing, with no relief in sight. Advances in production technologies (especially micro-electronics) lead to shorter product development times, hence sharper competition to innovate. Moreover, in the technically more advanced societies, there seems to be a growing dissatisfaction with simple materialist economics, resulting in growing pressures from labor, consumer groups, regulatory agencies, etc. (Schuhmacher 1973, Toffler 1980).

Advances in communications technologies expand the scope of business markets into other countries and cultures. Entrepreneurs in Hong Kong and London compete for sales in Minneapolis. The Third World rumbles for attention. While the advance of production and communications

* Discussion group no. 4, at the IFIP/IIASA Task Force Meeting on Processes and Tools for Decision Support, Schloss Laxenburg, Austria. Proceedings as Fick and Sprague (1980). Members of the group were P. Boxer, F. Flores, R. Hackathorn, P. Hearson, G. Kochetkov, S. Persson, C. Stabell, B. Trippett

technologies makes the manager's world bigger, more complicated and more dynamic, there do not seem to be compensating technological contributions for the task of management itself.

The medium of management is almost entirely linguistic. Computers, as symbol processors, would seem to be the ideal cognitive complement, extending the manager's memory and inferential abilities. The difficulties are certainly no longer for want of information processing technology. Newspapers and trade journals are screaming with new computer products that are faster, cheaper and more powerful than those the month before. But these are all improvements in the *form* of information processing. The more basic problem, we believe, is in the *content* of what computers do for management.

This book is motivated by what is seen as a gap between computer science and its areas of application, in particular to the management of organizations. The literature that describes what managers do and how they might do it better tends to be heavily sociological. Computer people tend to regard this somewhat impatiently, finding it 'soft', and so prefer their own more tractable -- albeit considerably less realistic -- models of management. Rather patronizingly, systems are designed to be able to interact with the 'naive user'. What is meant, of course, is the technically inexperienced user, though the associations are often generalized.

The communications gap between computer science and management is seen especially in their respective views of data. To the computer scientist, data is composed of bits, has such types as character strings, numbers (octal, hexadecimal, floating point, etc.), and can be collected in 'data structures' such as lists, arrays, records, relations, etc. To the manager, data is distinguished in terms of its content: production data, accounting data, market research data. The views of course reflect different pre-occupations. The two perspectives must merge, however, in order to achieve a successful computer application. Unfortunately, there is little science at this meeting point.

Applications are analyzed on a one by one basis with little generalization of these experiences between applications or between organizations. The recognized problems are concerned mainly with software compatibility and other technical issues. On the other hand, the fundamental problem is often not that the system 'works right', but rather that it does the 'right things'. There is hardly any research about the *content* of computer applications.

The two areas most relevant to this problem are the research in database semantics' within the field of database management (DM) and the knowledge representation work in artificial intelligence (AI). In both cases, however, the representations proposed are general purpose, making no presumption of organizational (or any other kinds of) application areas. The result is that the use of these representations is ad hoc and idiosyncratic. There is no accumulation of knowledge from one application to another. We are continually starting over from scratch.

The goal here is to initiate research into computational representations that *are* domain dependent. The focus is therefore not on data generally, but on *management data*. Clearly, management data cannot be distinguished simply by its symbolic appearance. What is relevant is the *semantics* of management data, what it refers to in the organization, and what commonalities can be found between various administrative applications.

Given the thousands and thousands of organizations that exist throughout the world, public and private, in manufacturing, financial, regulatory or other service areas, this objective may sound rather preposterous. But, the goal is not to solve all the world's problems. It is rather to identify a course of technology development that doesn't repeatedly throw away what it learns.

That there is something common to management in all these different organizations is evidenced by the fact that numerous management schools exist and flourish. In the U.S., thousands of MBA's (= Master of Business Administration) are graduated each year. Introduction of management training is further seen as an important component of aid programs to developing countries.

Much of what is taught in management schools is not formal knowledge, however. Aside from operations research and statistics, management courses also include organizational psychology, marketing, business policy and so on. These are typically taught using case studies. Accounting and finance might be regarded as areas of intermediate formalization. It is a serious issue whether the more behavioral topics of management studies can *ever* be formalized. In any case, there is no rigorous theory in sight at present. On the other hand, the more formal topics do also play an important role in management training. The synthesis of these 'hard' and 'soft' subjects is a central issue to management education. It is likewise the basic issue for developing effective management decision support systems.

But what is meant by saying that some management knowledge is 'hard', some 'soft'? What makes some managerial problems 'structured' others 'unstructured'? The basic issues are epistemological. A major part of how managers know their world is through linguistic inputs in the form of printed reports, correspondence, conversations, discussions in meetings, etc. The data maintained by a computer system constitutes one part of this larger organizational language. Our strategy for evaluating the potential contribution of information technology in managerial tasks is therefore to look at the (actual, possible) role of formalized data in management activities. To find the areas where information technology can contribute, we need to have some way of mapping out the domain of managerial knowledge, and what parts of that might be augmented computationally.

B. DEFINITIONS: SEMANTICS, ONTOLOGY, EPISTEMOLOGY

These are three terms that occur repeatedly throughout the book. Since they are fundamental to the discussion, starting definitions are warranted.

Semantics is the relationship between the vocabulary and expressions of a language and the things or phenomena signified. It is the mapping between language (or data) and the world.

Ontology asks the question, "what is there?". It is the set of assumptions of what basic entities exist in the world, quite apart from how we describe them in language. For instance, it is common to assume an ontology including physical objects. Of interest is whether these are the only things we need to assume in order to explain managerial cognition.

Epistemology asks the question "how do you know?". It is concerned with the foundations of knowledge. Whereas ontology hypothesizes basic entities, and semantics relates those entities to language, epistemology is what we do with the language in constructing higher order concepts. An epistemology of management therefore seeks to find the principle conceptual structures used by managers.

A fourth term, which we tend to use somewhat more loosely, is cognition. Cognition refers to mental activity. However, we also apply this to describe the symbolic activity of machines insofar as it behaviorally resembles human cognition.

C. NATURAL VS FORMAL LANGUAGE IN ORGANIZATIONS

Sometimes in a discussion an apparent disagreement ends up to be only a difference in semantics. The parties attached different meanings to the terms they were using. This book is about similar types of semantic problems raised to the level of communication and discussion throughout the organization. Our particular concern is the role that information technology might play in these processes.

This involves, in particular, technologies to store and manipulate the organization's data, namely database management systems. Our interest here is not with the technical details of these systems, but rather their effect on the organization and its management. Of special concern is the role databases play as a communications channel between separated parties in the organization. How do these parties know to attach the same meaning to the data they find in the database?

The problem of semantics in communication is of course an old one and has been the object of considerable linguistic and philosophical study. While current theories appear to be making progress, many deep problems remain. These studies apply to all uses of language, however, and therefore have to deal with the immense variation of all aspects of human experience, from baby-talk to poetry. Our working hypothesis is that the language of administration, especially those communications likely to be routed through information systems, are more restricted, hence more tractable. Managers of course converse using natural language. The language is 'natural' in the sense that it is a product of cultural evolution (Whorf, 1956). Contrasting with natural languages are artificial or *formal* languages where the syntax and semantics are specified in fixed and exacting rules. The temptation is to distinguish natural from formal languages on the basis of syntactic complexity and/or semantic range. This however would be relative to the state of the art in linguistics, which is advancing on both counts. (See, for example, the claims of Montague, 1974, regarding "English as a Formal Language".) The distinction we emphasize is, rather, one of *authority* -- the syntax and semantics of natural languages is decided by the linguistic population as a whole (more often perhaps by evolving accident than consciously negotiated consensus). Formal languages, whose character is embodied in explicit rules, are the product of a single authority, whose pronouncements remain fixed. Hence, though we might conceive of a set of explicit rules explaining the structure and scope of English, this will (here) still be a formal language since it is then fixed by the rules.

The distinction between natural and formal languages is a recurring theme in this book. While an information system might standardize the vocabulary and form of the communications routed through it, the system does not control the meanings users attach to the symbols that are communicated. That is to say, the system enforces syntax but not semantics. Thus, a basic issue is how do users of an information system, separated in space and time, know what the other is communicating about?

The linguistic/philosophical research on natural language semantics will obviously be of use here. However, that work is mainly directed towards explaining language phenomena that are otherwise regarded as beyond any particular authority's control to modify.

However, in information systems we do control the syntax and vocabulary and (partly, potentially) the way this language is taught to its users. Thus, the semantics of communication through an information system is more a matter of design and deliberated consensus.

As noted above, databases are regarded here as a convenient focal point for studying this issue. The data, whether routed through electronic networks or communicated through I/O devices, relies on the basic logical structure and definitions of the database(s) that they access. The semantics of a database is the correspondence between its symbolic data representations (a formal language) and phenomena in the organizational and/or societal environment. Our interest will be to explore the nature of this correspondence and how it arises, whether naturally or by design.

D. DATABASES AS FORMAL LANGUAGE ASSERTIONS

The above definition of semantics was of course informal and introductory. This concept will be developed more carefully in later chapters. However, for the moment it suffices to regard database semantics as the relationship between 'data and reality' (Kent 1978).

Semantic issues have been a recurring theme in the database management literature.* Much of this discussion is concerned with developing richer, more expressive data models. This is a very worthwhile enterprise, but it is not semantics in the sense meant here. It is rather, the design of syntactically richer, hence more expressive languages.

Here we need to be careful with terminology. In ordinary language we classify sentences to be declarative, interrogative, imperative, etc. Suppose we regard a database as a collection of declarative statements about the organizational environment. Let the language in which these statements are expressed be called L. Then the syntax of L, that is, its basic vocabulary and compound expressions, are determined by the database *schema*. The *data model* is the set of representational constructs for defining the schema.

The data model is then a sort of meta-language** for describing the syntax (structure, permissible vocabulary) of the assertions made in the

---

* For instance, Abrial (1974), Chen (1976), Smith and Smith (1977), Biller and Neuhold (1978), Lee and Gerritsen (1978), Codd (1979), Mylopoulos, Bernstein and Wong (1980), Hammer and McLeod (1981), Brodie (1982), Griffith (1982).

** There is a technical distinction between the data model, which is a collection of representational constructs, and a *data description language* (DDL), which is used to specify the schema. The data model is more conceptual, having a variety of notations (e.g., data structure diagrams). The DDL is a particular notation for the data model, possibly specifying other implementation aspects as well. This distinction is however not material to the discussion here.

database. This is not a typical characterization. Databases are normally not regarded as languages in themselves but more often as tabular arrangements of data items, e.g., as in the relational data model of Codd (1970).

The view of databases as *collections of assertions in a formal language* underlies all of the observations we have to make in this book. This apparently innocuous change in perspective leads to some substantially different issues about databases than have so far been considered. As will be seen, these are linked with foundational aspects in other areas such as organizational theory, accounting, and management decision making.

E. THE PROMISE OF ARTIFICIAL INTELLIGENCE

Another theme throughout this book will be the potential role of artificial intelligence (AI) in management applications. Whereas database management (DM) has always had a strong pragmatic orientation and was realized in products and applications from the outset, AI has had a different history. The original concerns of AI were in using the computer as a theoretical model in psychology (Newell and Simon 1972, Simon 1969/1981, Simon 1981a).

This is still an important theme. For example, much of the work on semantic nets has had the goal of modeling the associative structures of human memory (Quillian 1968, Norman and Rumelhart 1975, Findler 1979). Using the machine to model human cognition leads also to attempts to extend cognition. This was the goal of the early project MAC ('machine aided cognition') at MIT. However, in this agenda, AI seemed to be less successful. For a long time it seemed to produce only 'toy systems' that served to illustrate some theoretical point or another, but were not otherwise extendable.

More recently, though, AI has begun to break out of its ivory tower image and has started to win market appeal. Various AI-based companies are springing up. The principle areas of commercial interest seem to be in robotics and so-called expert systems.

The interest in robotics is mainly for industrial applications, especially where the work requires high precision (e.g. electronic circuitry), is tedious (various types of assembly line work), or dangerous (as in atomic plants). The advantage over conventional mechanization is that robots are teachable (programmable) to follow prototype human behavior of the task.* Thus they are the ideal of Taylor's (1911) 'scientific management', a mechanistic conception of labor management that was popular, but unsuccessful, during the first part of the century.

Expert systems are the intellectual counterpart to robots. Here the promise is to replicate the application of various types of professional knowledge, e.g. of medical doctors, lawyers, engineers. The economic motivations are however somewhat different. Professional training is an expensive and time consuming process. As the background knowledge required becomes more complex, a greater proportion of the individual's life is spent in training and less in the productive application of that knowledge. Further, this intellectual investment is lost when the individual dies, retires or changes professions. Also, expertise of this sort

* The basic research issues are the coordination of motor devices with tactile and visual sensors. Kent (1981) discusses the neuro-physiological comparisons.

tends to be inefficiently distributed geographically. For example, medical specialists tend to concentrate in the larger cities of wealthier countries. This leaves rural villages and poorer countries unattended.

The types of expertise embodied in expert systems are those that are 'rule based'; that is, they can be described in terms of fixed and explicit rules. A well-known example is the MYCIN system (Shortliffe 1976) for doing medical diagnosis.

The central problem in developing expert systems is so-called 'knowledge representation'. This has two aspects. One is the development of convenient yet robust formalisms for expressing expert knowledge in a way interpretable to the machine. Several alternatives are production systems (e.g., Davis and King 1975), semantic nets (e.g., Brachman 1979), logic programming (e.g., Kowalski 1979a). Brachman and Smith (1980) is a survey of ongoing research in knowledge representation around the world.

The other part of the problem, sometimes called 'knowledge engineering', is the application of these formalisms to a particular problem domain. The most successful applications have been to medical areas, but include other scientific domains such as analytical chemistry, synthetic organic chemistry, protein X-ray crystallography, biochemistry, cognitive psychology and geological prospecting (Infotech 1981).

There is, on the other hand, some doubt as to the sufficiency of these approaches to fully represent what might be called 'mature expertise', i.e., that gained not simply from formal training, but refined through long experience. S. Dreyfus (1982) regards rule based knowledge as merely an early stage in the formation of mature expertise, which is more holistic and integrated in character. He uses the example of learning to drive a car. One begins with a certain set of learned rules – e.g. at what speed to shift gears, when to turn the wheel in parallel parking. Later these rules are refined for unusual circumstances, e.g. shifting gears on hills or curves, parallel parking on an incline. However, still later, the awareness of distinct rules fades entirely and we simply shift, park, etc. as 'second nature'.

The expertise of a mature doctor or engineer seems also to have this holistic character. (Lee 1982 examines some of the social consequences of this conjecture.) Of interest here is whether knowledge engineering approaches are applicable to management. A problem seems to be that managers apparently rely considerably less than doctors or engineers on a formalized body of knowledge. We return to this in the final chapter.

F. MANAGEMENT DECISION SUPPORT SYSTEMS

Databases are typically regarded as the central component of information systems. Other components included data communications, transaction processing routines, user interfaces, etc. In the 1960's and early 70's these used to be called Management Information Systems (MIS). The implication was that they were primarily directed towards the information needs of managers. However the subsequent experience has been that these systems have concentrated mainly on operational level data processing, and have had relatively little impact on management activity.

Gorry and Scott-Morton (1971) explain this in terms of the 'unstructuredness' of management tasks. Making use of a popular taxonomy by Anthony (1965), management activity is distinguished as:

a) strategic planning

b) management control

c) operational control

Operational control involves managing the productive operations of the organization. It is task oriented and involves planning periods measured in days or weeks. Management control involves intermediating between higher level planning and the operational level. Importantly, it is not merely a vertical link, but involves substantial horizontal coordination between the various functional departments. Strategic planning involves the positioning of the firm with respect to markets and competition. It is outward directed, and has long term planning horizons.

Gorry and Scott-Morton also make use of a distinction by Simon(1960/1977) between 'programmed' vs 'non-programmed' tasks. A programmed task is one where a decision algorithm or procedure exists. To avoid associations with computer programs, they change the terminology to 'structured' vs 'non-structured'.* Their general observation is that tasks at the operational level tend to be much more structured than those of the upper levels. (See Figure [1.1].) Managers at the Management Control level tend to be exception handlers. They deal with the shortcomings of plans, the surprise changes, etc. Routinized activities are delegated to subordinates, thus their task remains highly unstructured. Further, in their capacity as coordinators, they rely heavily on diplomatic skills: effectively arbitrating between diverse personalities is a highly unstructured activity.

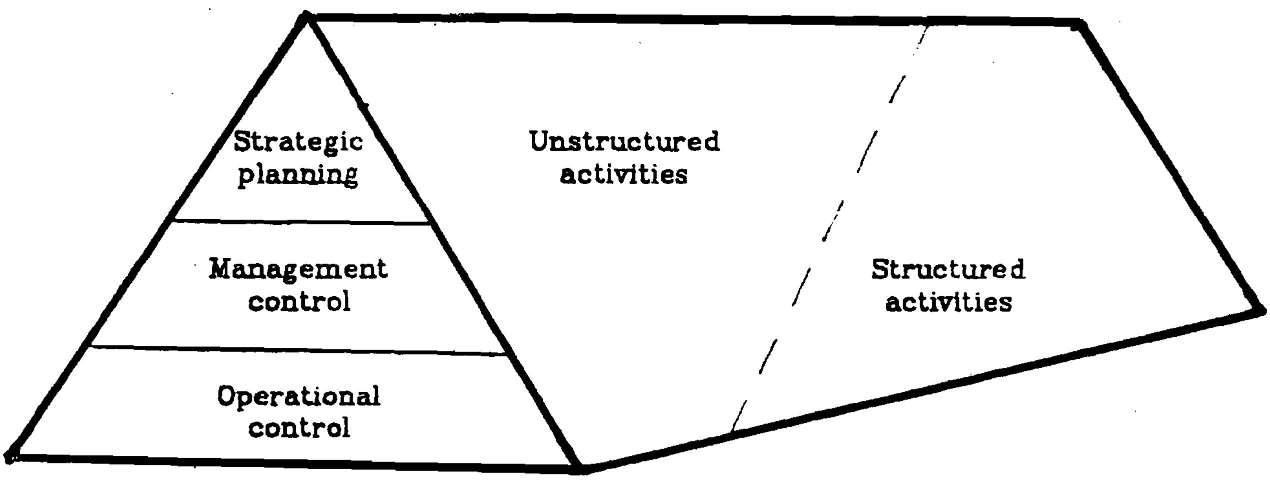


Figure 1.1. Structured vs unstructured tasks compared to management levels.

* This is unfortunate since the term 'structured' is much more ambiguous than 'programmed'. Importantly, Simon's concept of 'programmed' tasks referred to the state of rationalization in the organization. 'Structured' is often used to describe an inherent property of the task itself, rather than the organization's policy towards it.

At the strategic planning level these same characteristics apply to an even greater degree. The environments they face are primarily social ones -- economic and political movements, competitive behavior, labor complaints, market trends. There is little that formal science has to contribute in these areas.

If information technology is to have an impact at these levels of management, a different set of starting assumptions is needed. This has led to the concept of a management decision support system (DSS). Keen and Scott-Morton (1978), often regarded as definitional authorities in this area, remark:

> Decision Support Systems (DSS) represent a point of view on the role of the computer in the management decision-making process. Decision support implies the use of computers to:
>
> 1. Assist managers in their decision processes in semistructured tasks.
> 2. Support, rather than replace, managerial judgment.
> 3. Improve the effectiveness of decision-making rather than its efficiency.

We note the emphasis on decision. That is perhaps already too narrow a focus insofar as it connotes a final choice between well-defined alternatives. Instead we prefer a broader conception of managerial cognition that includes aspects of reality testing, problem finding, scenario generation and just plain muddling through.

Also, the above characterization of DSS seems to presume a solitary decision maker (this is true of expert systems as well). Managers however spend relatively little of their time in solitary contemplation (22 per cent in Mintzberg's 1973 study). Organizations are social activities and so too are the activities of its managers.

Here we see the interplay between concepts of information systems and decision support systems. Information systems constitute a network of formal communications in the organization. Decision support systems will need to interact with this in some more or less 'loosely coupled' way.

The term 'decision support' has now become well established. It is perhaps more appropriately characterized as a philosophical attitude towards technology application than a technology itself. In that regard, the continuation of the remarks by the discussion group quoted at the beginning are noteworthy:

> We believe that managers live in a constant state of transition. Perplexity is always within the manager's mind, and this will not change. The manager will continue to act without fully understanding and will not consider this to be a problem; while attempting to increase his understanding, he never expects to arrive at a full understanding.*

* op. cit. discussion group no. 4.

G. LANGUAGE AND COGNITION IN ORGANIZATIONS

The concepts of language and cognition are strongly related. While we do not necessarily think in language, it is dubious whether very many products of thought, i.e. our culture, could exist without language.

The relationship between language and managerial cognition has its counterpart in computer technology. The theory of computers, so-called ***automata theory***, postulates a hierarchy of abstract machines of increasing computational power. Hopcroft and Ullman (1969) show that the concept of an automaton is equivalent to a grammar for translating an input set of symbols to an output set. The vocabulary of symbols constitutes a formal language processed by the grammar, and the various abstract automata are distinguishable based on the syntactic complexity of the formal languages they process. These are compared to the categories of formal languages proposed by Chomsky (see e.g. Levelt 1974). The correspondence is as follows:

GRAMMAR		AUTOMATON
3. REGULAR	=	FINITE
2. CONTEXT FREE	=	PUSHDOWN
1. CONTEXT SENSITIVE	=	LINEAR BOUNDED
0. RECURSIVELY ENUMERABLE	=	TURING MACHINE

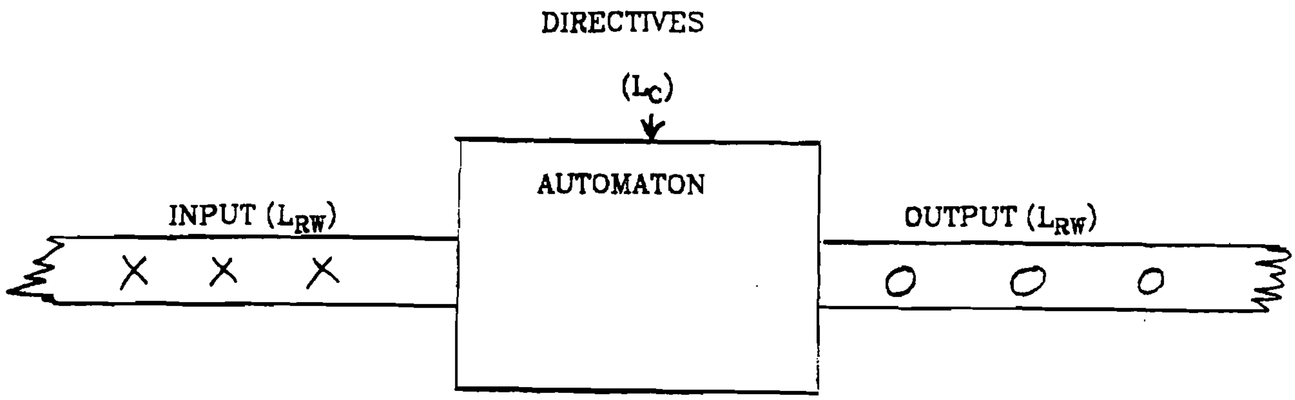
The recent work in Artificial Intelligence suggests a generalization of the concept of cognition to apply not only to humans but also to machines. Adopting this usage, we see a certain parallel between human vs mechanical cognition and the earlier distinction made between natural vs formal languages.

The relationships are shown in Figure [1.2]. The diagram has two parts, comparing managers to an automaton. In part a) a standard conception of a Turing machine is drawn having an input, output and intermediate store. The symbol stream of this automaton is presumed to constitute a formal language, L_{RW}, describing the real world. L_{RW} therefore corresponds to the data stored, updated and retrieved in the organization's databases. This language is distinguished from L_C, the computer language for programming the automaton.

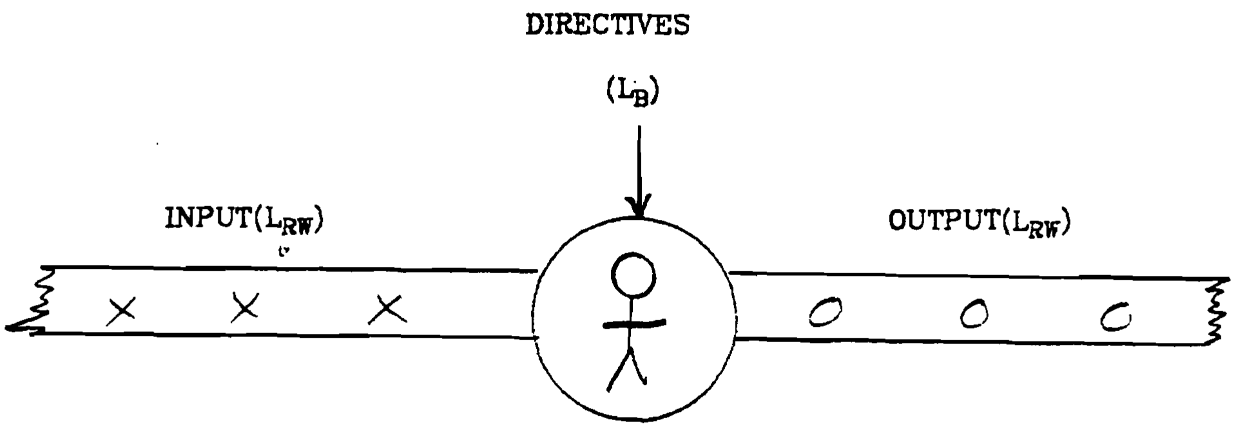
In part b) of Figure [1.2], the role of the automaton is substituted by a human manager, or perhaps a team of managers. Again, there is a language, L_{RW}, describing the real world, which these managers process. In this case L_{RW} is a natural language, though it may contain formal language components. (Recall that formal languages have explicit and fixed rules controlling their syntax and semantics. Natural languages may evolve, depending on the consensus of the linguistic community.)

Corresponding to L_C, the computer language used to program the automaton, managers learn their duties in L_B (for 'bureaucracy'), the language of their job descriptions and other directives.

The vocabulary of L_B and L_{RW} may of course overlap. Likewise, the vocabulary of L_C and L_{RW} may also overlap. (LISP programmers, for



a. Mechanical cognition



b. Managerial cognition

Figure *[1.2]*.

instance, regularly deny the difference between programs and data.) The distinctions made here are for expository purposes, based mainly on semantic scope. L_C and L_B refer to the actions, respectively, of the automaton and the manager. L_{RW} refers to descriptions of the (actual, possible) environment that the automaton and manager process.

H. ORGANIZATION OF CHAPTERS

The distinctions between L_{RW}, describing the real world, and L_B vs L_C, used for instructing (human vs computer) information processors, provide the organizing basis for the chapters to follow.

Chapters [2] through [5] are concerned specifically with aspects of L_{RW}.

Chapter [2], [Databases and Logic], considers the use of databases for decision support purposes. Rather than simple retrieval of data, these applications require inferencing on the elementary facts to achieve the more abstracted, higher level concepts used by managers. This motivates an examination of ongoing developments relating database representations to predicate logic, and to the possibilities of so-called 'logic programming'.

Another motivation for relating databases to logic is to relate the substantial existing research on formal semantics to the role of data in organizations. This is the subject of chapter [3], [Formal Semantics of Databases]. Semantic issues are developed, in particular the semantic problems arising in dynamic environments: how to maintain a consistent interpretation of the symbolic vocabulary. This involves considerations of so-called 'possible worlds semantics'.

In Chapter [4], [Naming: Individuals and Natural Kinds], the philosophical literature on this problem is brought to bear. The issues of names for individual objects (proper names) are shown to be related to those for generic terms for 'natural kinds'. The concept of possible world is examined in the emergent 'new theory of reference', and the sociological aspects of semantics are observed. These aspects are particularly characteristic of language use in organizations.

Whereas chapters [3] and [4] deal with the semantics of qualitative terms identifying individuals and classes, chapter [5], [Measurement], considers numeric data. Current research in measurement theory is outlined. The semantic aspects of measurement are discussed.

In organizational administration, one particular type of measurement dominates, namely accounting measurement. In chapter [6], [The Semantics of Accounting], the cliche that 'accounting is the language of business' is taken literally. Accounting statements are examined as constituting a formal language, and the semantic foundations are studied. As measurement, accounting data consists of two components: the domain of measurement, i.e. the objects being measured, and the scale of measurement, i.e. monetary values. Among the objects that accounting measures, contractual objects figure prominently. Included here are receivables, payables, notes, bonds, stocks, leases, licenses, insurance contracts, etc. Indeed, contractual relationships are the binding force of the economy and of organizations themselves. They distinguish an organization from a mere collection of objects and people.

However, from a semantic standpoint, contracts are rather difficult things to understand. We seem to treat them simultaneously as *relationships* and *objects*. In chapter [7], [The Logical Structure of Contracts], an informal discussion of these issues is presented. (A more rigorous treatment is postponed to the appendix.) Fundamental insights are provided by the developing area of *deontic logic*. 'Deontic' refers to a concern with normative systems. Efforts to formalize reasoning in this area include logical operators of obligation, permission and prohibition. These operators apply to human *actions*. Thus an explication of action is also required. This includes a concept of personal *responsibility* for a change in the state of the world. From these components, a concept of contractual commitment is constructed. Various problems remain. One is the extension of propositional deontic logics to recognize first order individuals. Another is the treatment of *contingent* commitment, currently at the center of debate in deontic logic. Further, the role of time is fundamental to contractual commitment, requiring the integration of temporal logic with deontic logic.

Referring back to Figure [1.2], the chapters thus far have dealt primarily with L_{RW}, the (formal and informal) language describing the organizational environment. Attention next turns to the way this language is processed, i.e. by machines and people in the organization. Rather than confront the mechanical details of technology development and the many unanswered questions of cognitive psychology, we instead examine the imperative languages (L_C and L_B) by which information processing in the organization is directed.

In chapter [6], [Analyzing Red Tape: Deontic Performatives], the discussion relating to contractual commitment ***between*** organizations is applied to analyze activity ***within*** the organization. The phenomena studied here are internal transactions and in particular the role of bureaucratic documents. Transactions that are merely informative (L_{RW}) are distinguished from those that are performative (L_B). The latter have an imperative aspect, involving the exercise of authority. These are shown to have an underlying structure analogous to contractual relationships, hence also relying on deontic aspects. An important difference between these transactions and purely informative ones is in the individuation of the document itself. It is by means of deontic performatives that the organization controls its human based information processing. A concept of bureaucratic software is suggested.

In chapter [9], [Bureaucracies, Bureaucrats and Information Technology], the comparison between human-based vs computer-based information processing in organizations is further considered. The neutral concept of bureaucracy originally proposed by Weber is contrasted with the negative connotations it has since acquired. The problem of bureaucratic rigidity is examined in terms of the personal interests of bureaucrats themselves, and the complexity of bureaucratic rule systems. A taxonomy by J. Galbraith is proposed for examining the effectiveness of different administrative methods for coping with complexity and uncertainty in the environment. Bureaucracy is observed to be effective for complex but stable (certain, predictable) environments. To cope with greater uncertainty, the organization needs to rely on greater discretion among its employees. An explanation for organizations that apparently cope well in environments that are both complex and uncertain is provided by the concept of 'corporate culture' by Deal and Kennedy.

In chapter [10], [Applications Software and Organizational Change], the problems of software adaptation are examined from an organizational standpoint. This is currently a critical issue for the software industry. The difficulties of software change limit the extent to which the organization can rely on computer technology for its administrative operations. Otherwise the technology, while efficient, will restrict the organization's ability to adapt and innovate. The effect is similar to the inflexibilities of bureaucratic rationalization, but the causes are different. Innovations from artificial intelligence are proposed to relieve this problem.

Artificial intelligence won't, however, provide a complete answer. As regards organizational management, human judgment and understanding will continue to be needed. This is not moralizing, rather epistemologizing. In chapter [11], [Towards a Theory of Management Decision Support], the arguments of the previous chapters are reviewed and integrated into an evaluation of the potential impact of future information technology on management. Here the risky game of technology forecasting is avoided by considering a logical idealization of computer based information processing, namely as a formal language processor.

The basic limitation is semantic change in the language L_{RW} describing the environment. The phenomena in the world are far richer in their number and aspects than our vocabulary for describing them. In natural language we circumvent this difficulty by changing the semantics as we speak. These are not the long term gradual developments usually studied in socio-linguistics, but the temporary shifts in meaning we introduce into each conversation where a new idea is discussed. As the organization seeks to adapt and innovate, these temporary shifts in meaning become part of the organization's language. Computers, as automata, control only the syntax (= vocabulary + formation rules + transformation rules) of the formal languages they process. The semantics of these languages (i.e. L_{RW}) is not part of the computer system itself, but rather of the way it is interpreted by its users. The inferences performed by the system are valid only under the presumption of a fixed semantics. Even in an idealized form of the technology, this will not change*. Herein lies the kernal of a theoretical foundation for management support systems. The mechanisms of semantic change, whether local to a single conversation, a change in the organizational dialect, or widespread change in the language of the whole society, depends on sociological mechanisms.

The role of computer aids in management processes will therefore be limited to situations, narrow or broad, short term or long term, where the semantics can be assumed fixed. The complementary role of human cognition will be to track and/or initiate semantic change and delineate the contexts where the semantics can be assumed stable, hence where computational inference will be applicable.

The consequences of these observations for developing organization-wide information systems is to emphasize the restrictive effect these systems can have on the organization's ability to adapt and innovate. Improvements in the modifiability of software are a critical consideration.

Likewise, these same observations provide a thoretical foundation for a separate field of decision support systems, distinct both from information systems and artificial intelligence.

Throughout the other chapters, three senses of the term 'model' are used: a ***data model***, i.e. a descriptive representation scheme; an ***inferential model*** (as in operations research or artificial intelligence), where computational manipulations are meant to correspond to structural characteristics of the world; and a ***semantic model***, in the Tarskian sense of an ***interpretation*** of a formal system, e.g. of a data model or inferential

* Fully automated factories notwithstanding. A robot factory still has the problem of marketing to changing consumer tastes.

model. It is this third sense of model that is emphasized in decision support systems.

Decision support systems are concerned with models management in all three of these senses. We need to manage descriptive representations, inferencing schemes on these representations, but also the interpretations applied to our models. The consequences are not only for technology development but also for management education. Managers need to be sensitive to what a computational model can and cannot do for them. The limitations are semantic. Indeed, these are the limitations of organizational rationalization in general. Just where and how managers should manage this rationalization is the central theoretical issue for further DSS research.

CHAPTER 2:

DATABASES AND LOGIC

CONTENTS

A. DATABASE MANAGEMENT

Database management (DM) arose originally from a need for a specialization of labor in data processing. Applications programmers had the dual function of satisfying user requirements as well as efficiently maintaining the data on various storage devices.

As long as applications tended to be relatively independent, this was not a great problem. However, as more and more data files came to be shared among various applications, coordination problems arose. Different applications favored different types of data organization.

Database Management Systems (DBMSs) offered a separation of these concerns. Essentially, a DBMS translates between an abstracted view of data, accessed by application programs, and its actual physical representation.* What the appropriate abstracted view should be, so-called 'data models', became an interesting research question and has been the subject of prolonged debate for nearly a decade. The basic camps, eventually, centered around a graphical view called the Network Model as opposed to a tabular view, the Relational Model. (Date, 1977, gives a good comparison.) While the two views are closely compatible, the Network Model seems to have certain advantages from the user engineering standpoint, and has been more widely implemented. The Relational Model, on the other hand, is mathematically simpler, and for that reason has been the more favored view in research discussions. The Relational Model is also adopted here as representing the database management paradigm.

* The abstraction process may actually go a step further as recommended by the ANSI/X3/SPARC report (Tsichritsis and Klug, 1977). Following that report, programs would access an 'external view' of the data, which is a subset of a master view called the 'conceptual schema'. This in turn is mapped to the 'internal schema' indicating actual physical storage.

B. THE RELATIONAL DATA MODEL

The Relational Model was originally proposed by Codd (1970). In this view, data items are regarded as arranged in rectangular tables consisting of columns and rows. Columns are called ***attributes***, rows are called ***tuples***, while the entire table is called a ***relation***. An example relation, containing data on employees, is the following:

EMPLOYEE	(ID#,	NAME,	RANK,	SALARY)
	12	JONES	CLERK	10000
	51	SMITH	CLERK	10000
	27	DOE	MANAGER	25000
	05	ELIOT	PRESIDENT	50000

Note that rows correspond to individual employees whereas the columns indicate the various recorded features of the employee. This is the general convention, i.e. that rows correspond to individuals in the environment ('instances') while columns indicate their attributes. In the EMPLOYEE relation, the attribute ID# (identification number) is a 'key attribute', that is, a unique identifier (of the individual in the environment corresponding to the tuple). Such keys serve as cross references to other relations, such as in the following relation, showing superior/subordinate relationships.

WORKS-FOR	(SUPERIOR#,	SUBORDINATE#)
	27	12
	27	51
	05	27

In this case, both SUPERIOR# and SUBORDINATE# refer to ID# data items in the EMPLOYEE relationship. The identifying key for the WORKS-FOR relation is however the conjunct of the SUPERIOR# and SUBORDINATE# attributes.

In the theory behind the Relational Model, database relations are regarded as mathematical relations over various domains of data items. An important concept in this theory is the so-called 'functional dependency' that may arise between attribute domains. That is, if one attribute, A, is functionally dependent on another, B, then an update to B requires a corresponding update to A.

In the above example, for instance, it may be the case that salary depends on rank. That is, each rank has a fixed salary. Hence, knowing an employee's rank, we can determine his or her salary. In this case, the database would be redundant, since the salaries of clerks are recorded twice. To avoid potential inconsistencies (e.g. having one clerk's salary different than another's) the database should be normalized so that each such fact is recorded only once. In this example, the EMPLOYEE relation would be divided into two relations, EMPLOYEE and PAY-SCALE, as shown below. (For further discussion on normalization, see Codd, 1972, Fagin, 1977.) Note that in the PAY-SCALE relation, the attribute RANK serves as the identifying key.

EMPLOYEE	(ID#,	NAME,	RANK)
	12	JONES	CLERK
	51	SMITH	CLERK
	27	DOE	MANAGER
	05	ELIOT	PRESIDENT

PAY-SCALE	(RANK,	SALARY)
	CLERK	10000
	MANAGER	25000
	PRESIDENT	50000

However, this decomposition is appropriate only *if* the organization's personnel policy makes salary a unique function of rank. The equal salaries of the two clerks may only have been an accidental coincidence, not due to a functional dependency. This is a fundamental point: functional dependencies cannot be detected from patterns in the actual data alone. They reflect relationships between *possible* values of attributes.

This is due to the fact that organizational databases are dynamic, that is, they are continually being updated reflecting the effect of organizational transactions such as sales, inter-departmental transfers, production runs, etc. If the database were completely static, functional dependencies could be detected from the actual data, but then they would not be of interest; since there are no updates, no accidental inconsistencies could arise.

C. INFERENCING ON DATABASES

The major use of DM databases to date has been in data processing applications; hence mainly for structured, operational level activities such as sales order processing, billing and inventory control. These applications are characterized by high volumes of routine transactions. Performance criteria are mainly speed and efficiency. Databases might also be useful in less structured, longer range activities, though the requirements in this case are somewhat different:

a. information is usually required in more summarized form

b. access is less routine — information must be retrievable in a variety of forms and combinations

c. the information is often used in combination with other informational and computational resources.

These are criteria for using DM databases in *decision support* applications. The principle point is that the data needs in these cases, though contained in the database, will often not be at the detail level nor in the structural arrangement in which the database was designed. It is for these uses that a mechanism providing inferencing on the database is needed.

One obvious way of summarizing data is simple arithmetic calculations — e.g. counts of inventory. Lacking however is a corresponding framework of qualitative inferencing. For instance, if you have an inventory of three apples and two oranges and count them up, you have five 'things', but what descriptive label should be attached to this broader class? In this case a system of qualitative inference is needed. More realistic examples abound, e.g. in accounting data if you have $500 in cash and $700 in accounts receivable, then you have $1,200, but of what? Conversely, one might wish to make a query about the quick assets of the company when the database only contained data on cash and accounts receivable.

D. PREDICATE CALCULUS AND LOGIC PROGRAMMING

Further discussion of database inferencing for decision support applications requires a brief background on predicate logic and its computational counterpart, logic programming.

1. Predicate Calculus

It is assumed that the reader is at least generally familiar with the first order predicate calculus (FOPC) and its syntax. The following is thus only a review.

The description of a logical system begins by declaring its *universe of discourse*. In a propositional (zero order) logic, this amounts to a set of statements (propositions) asserted to be true. In a first order logic, a separation is made between individual entities (or just *individuals*), and the properties and relationships to other individuals. The latter are indicated, respectively, by one and n-place *predicates*. For a first order logic the domain of discourse is called the *domain of individuals*. (For the moment, the individuals described by the logic can be imagined as discrete physical objects at a point in time.) In summary form, the basic constructs of a first order predicate calculus are as follows:

1. *Propositions*.

   These are complete logical statements having a truth value. These are indicated symbolically by capital letters —e.g. P,Q,R.

2. *Logical connectives*.

   These combine one or more propositions to form new logical statements, also having a truth value. The logical connectives used here are as follows:

↔	equivalence
→	implication
&	conjunction
V	disjunction (inclusive)
W	disjunction (exclusive)
~	negation

3. *Individual constants and variables.*

These stand for objects in the domain of discourse — e.g. individual trucks or employees.

Individual constants are denoted as one or more upper case letters, possibly containing non-leading digits or hyphens; e.g. A, GEORGE, TRUCK-7.

Individual variables are denoted as either lower case letters, e.g. x, y, z, or as a "?" followed by one or more capital letters or digits, e.g. ?ID, ?SALARY. (The dual notation here is a compromise between the logical convention of variables as lower case letters, and the database management convention of capitalizing names of attributes that are recognized as variables in a logical interpretation.)

4. *Functions.*

These map one or more individuals to another — e.g. SUPERVISOR (JONES) refers to another individual who is Jones' supervisor. Functions may take zero or more arguments and always result in a reference to a single individual. Functions may thus appear wherever an individual constant is allowed. Indeed, a zero-place function is the same as an individual constant. Functions are therefore denoted in the same way as individual constants, but followed by an argument list, e.g, F(A), BOSS(SMITH)

5. *Predicates.*

   These indicate features, properties, attributes, etc., applied to zero or more individuals. Predicates will be denoted by upper case letters or words, e.g. P(x), RED(?X), OWN(x,y). When a predicate is applied to individual constants or to quantified individual variables (see below), or to functions of these, it has a truth value and may be combined to form other logical statements using the logical connectives above. A zero-place predicate is equivalent to a proposition.

6. *Logical quantifiers.*

   These indicate the range of individual variables. The principal ones are:

∀x	universal quantifier (for all x, for each x, — ranging over all individuals in the universe)
∃x	existential quantifier

(for some x —ranging over
at least one individual)

Parentheses are used in the usual fashion.

2. Logic Programming

Mechanical theorem proving in the predicate calculus has been a central area of AI research since its outset. As with logic generally, the original goal was to reproduce mathematical reasoning. Thus, an early success was the Logical Theorist program by Newell, Shaw and Simon (1963), which reproduced the proofs of Russell and Whitehead's *Principia Mathematica*. Indeed, the program found several original proofs of certain theorems. A more recent success is the AM* program of Lenat (Davis and Lenat, 1982). The goal in AM is not only to prove specified theorems from a given set of axioms, but also to decide for itself which axioms are interesting to prove. It thus is a model of mathematical discovery.

Just as modern logic is now used to formalize reasoning in non-mathematical subjects, AI theorem-proving systems have also been applied to model reasoning in other areas. Basic axioms about the world are asserted and the system deduces further statements (theorems) based on these axioms.

Whereas mechanical theorem-proving for the propositional calculus is relatively easy, theorem-proving in the (first order) predicate calculus

* Lenat: "the original meaning of this mnemonic has been abandoned. As Exodus states, 'I *AM* what I *AM*'." (Davis and Lenat, 1982, p. 3).

is computationally much more difficult. One problem is that there are typically a number of inference rules available, corresponding for example to different arrangements of leading quantifiers or different combinations of logical connectives. While these are a convenience to human logicians, they lead to excessive branching and an extremely large search space for mechanical proofs.

The so-called 'resolution method' of Robinson (1965) offers considerable computational simplification by reducing logical assertions to an elementary 'clausal' ('Horn clause') form. In this form, only one inference rule, resolution, is needed. (Resolution essentially combines the inference rules of modus ponens and substitution.) Assertions in clausal form have the following general pattern:

$$P_0 \leftarrow P_1 \,\&\, P_2 \,\&\, \ldots \,\&\, P_n.$$

where the P_i are predicates of the form $P(x_1, x_2, \ldots, x_k)$. This can be read: "to prove P_0 it is sufficient to prove P_1, P_2, ..., and P_n. All variables are assumed to be universally quantified. It can be shown* that any first order assertion can be reduced to this form. The resolution method provides the basis for a family of theorem-proving languages that together have come to be known as 'logic programming'. The best known among these is the language PROLOG (abbreviating PROgramming in LOGic), originally invented by Alain Colmerauer about 1970. Useful texts are Kowalski (1979a), Coelho, et al. (1980), and Clocksin and Mellish (1981). The discussion here is based mainly on PROLOG, with slight syntactic variants

* This reduction requires the inclusion of so-called Skolem functions, which take the role of existential quantification. These are not discussed here. Further discussion of clausal form is given in Nilsson, 1980, and Clocksin and Mellish, 1981.

to make it consistent with the preceding logical notation.

In logic programming, one typically distinguishes between *facts* and *rules*. A fact is a clause containing only the left hand side and no variables. For example,

MALE(DICK).

SIBLING(DICK, JANE).

are facts. Rules are clauses with expressions on both sides of the implication and containing variables. For example,

BROTHER(x, y) ← SIBLING(x, y) & MALE(x)

Disjunction is expressed using multiple rules. For example, BROTHER(x, y) can be proven in two ways, namely:

BROTHER(x, y) ← SIBLING(x, y) & MALE(x).

BROTHER(x, y) ← SIBLING(x, y) & MALE(y).

The first is the rule just discussed; the second allows for the reverse matching of arguments (because SIBLING is symmetric while BROTHER is not). Though this is the typical way of indicating disjunction in logic programming, for notational simplicity the connective, V, will sometimes be used. This is assumed to have lower priority than &. For instance,

BROTHER(x, y) ← SIBLING(x,y) & MALE(x) V SIBLING(y,x) & MALE(y).

is equivalent to:

BROTHER(x, y) ← (SIBLING(x,y) & MALE(x)) V (SIBLING(y,x) & MALE(y)).

Goal theorems (i.e. things to be proved) are denoted with a question mark, e.g. ,

BROTHER(DICK, JANE) ?

asks whether DICK is the brother of JANE. In this example the system would respond YES. Variables can also occur in goal theorems. In these cases the system's response is similar to that of database queries, namely, it returns all combinations of variable bindings that result in a provable theorem. For instance, the logic program:

MALE(DICK).

MALE(TOM).

MALE(HARRY).

MALE(x) ?

would respond:

x = DICK

x = TOM

x = HARRY

A slightly more complicated example is the following:

SIBLING(DICK, SALLY).

SIBLING(TOM, DICK).

SIBLING(HARRY, TOM).

SIBLING(x, z) ← SIBLING(x, y) & SIBLING(y, z).

The last rule indicates that the SIBLING relationship is transitive. Thus, the query, SIBLING(x, SALLY) ?

results in the response:

x = DICK

x = TOM

x = HARRY

Note that three levels of inferencing are involved here. The first is simply a match to the fact, SIBLING(DICK, SALLY). The second requires the inference that TOM is a SIBLING to DICK and that DICK is a SIBLING to SALLY so TOM and SALLY must be SIBLINGs. The third is similar but with the additional inference that HARRY is SIBLING to TOM so that HARRY must be a SIBLING to DICK, hence also SIBLING to SALLY.

An important aspect of logic programming as compared with other types of computer languages is that it is non-procedural, or 'declarative'. In purely declarative languages, the order in which statements are evaluated is not controlled by the programmer*. Thus the order of the statements in a logic program doesn't matter as regards the system's inferencing capability. (It may however make a difference from an efficiency standpoint.) Logic programs are therefore an extreme form of modularity in computer program design.

However, there is one aspect of this non-procedurality that has to be compromised in order to address practical applications; this is for numeric computations. To do calculations in a strictly logical way would involve inferencing on the basic axioms of arithmetic. This would be

* This is true of 'pure' logic programming. In PROLOG, a certain amount of execution control can be specified by using the so-called 'cut' operator.

impossibly inefficient for any but trivial numeric computations. Logic programs therefore make calls to special subroutines when arithmetic is done. This is denoted here using a functional notation plus the usual arithmetic operators (+, -, *, /) for addition, subtraction, multiplication and division. For example, consider the following logic program:

HEIGHT-IN-METERS(DICK, 1.5).

HEIGHT-IN-FEET(x, z) ⟵ HEIGHT-IN-METERS(x, y) & z = y * 3.28.

HEIGHT-IN-FEET(DICK, z) ?

z = 4.92.

Note that in logic programming, numeric constants are regarded as logical individuals. The subroutine invoked in computing z is logically regarded as a huge collection of facts giving all possible sums, products, etc.

What has just been described is the basic kernal of logic programming. Implementations include a variety of other aspects including in particular 'evaluable predicates' that have certain side effects permitting input/output, modification of assertions, etc. Also, more complex data structures (e.g. character strings, lists) are typically involved. These extensions enable logic programming to be used for a variety of applications beyond the usual conception of theorem-proving, e.g. natural language parsing, graph searches, user interfaces.

The motivation for introducing logic programming here is to examine the possibilities of database inferencing. This subject is considered next.

E. THE ENTITY-RELATIONSHIP INTERPRETATION

In the past decade, the Relational Model of Codd (1970) has clearly established the paradigm for database research. However, a criticism of the relational model is that it avoids commitment as to the semantics of the database, i.e. how the database structures signify or denote phenomena in the environment. A step in this direction is provided by the Entity-Relationship interpretation of Chen (1976). (This is normally called the Entity-Relationship *Model*, or ERM. It is, however, more an interpretation applied to the Relational Model.) The import of this approach is to draw attention to the role of relational keys. These are generally identifying labels for *entities* in the environment, e.g. part numbers, social security numbers. With this observation, certain relations serve to describe individual entities (entity relations), while others indicate relationships between entities (relationship relations).

The ERM highlights the *existential assumptions* of a database. Each tuple in a database is assumed to correspond to a particular entity in the environment or a relationship between entities.

The ERM is sometimes criticized that it fails to prescribe what count as entities, e.g. only physical objects? Should abstract objects also be admitted? The reply, of course, is that this depends on the organization's phenomenology. There is no absolute answer; what the organization recognizes as entities depends on its technology and view of the world. For instance, the popular example database

STUDENT (S#, ...)

COURSE (C#, ...)

ENROLLMENT (S#, C#, ...)

recognizes students and courses as entities, and enrollment as a relationship between them. This is a convenient view for university administrators, even though the concept of a 'course' is an abstraction that might be rather troublesome to pinpoint ontologically.

F. PREDICATE LOGIC INTERPRETATION

Ignoring, for the moment, the deeper semantic issues, the ERM has a straightforward interpretation in predicate logic:

a. entity relations = one-place predicates

b. relationship relations = multi-place predicates.

While this is a satisfactory interpretation of the definition of relations, the data in the relations are still unexplained. Generally, these seem to be of three types:

a. data items functioning as ***identifiers*** of entities (in the role of logical names)

b. data items corresponding to ***predicates***.

c. data items representing ***numeric measurements***.

For example, consider the relation:

EMPLOYEE	(NAME,	SEX,	SALARY)
	SMITH	MALE	35000
	JONES	FEMALE	42000

corresponding logical assertions would be:

EMPLOYEE(SMITH) & MALE(SMITH) & SALARY(SMITH, 35000).

EMPLOYEE(JONES) & FEMALE(JONES) & SALARY(JONES, 42000).

Here the values of the first attribute, NAME, translate as individual names in logic. The values of the second attribute, SEX, translate as predicate names, i.e. MALE(x), FEMALE(x). The values of the third attribute translate as numbers, which in logic programming are taken to be another type of individual. To relate the human individual to the numeric individual, a two place predicate, SALARY(x, n), is introduced. Since the use of numbers in databases typically indicates a functional mapping from the real world entity to a numeric domain, a functional notation is often used, e.g.

SALARY(SMITH) = 35000.

SALARY(JONES) = 42000.

Database management models typically distinguish between the ***structure*** and ***contents*** of the database. In the logical form this distinction is not made. In database management, the structure/content distinction gives rise to the view of databases as repositories, somewhat akin to physical inventories. A database ***query*** specifies retrieval conditions, and the database contents that match these conditions are delivered to the user. In logical form queries are processed not simply by matching character strings, but rather by logical inference (this point is elaborated below).

This reflects a fundamental difference in the two perspectives. Database management regards data as character strings that the system stores and delivers to the user upon request. The ***interpretation*** of these character strings lies outside the theoretical concern. (Recall: GIGO = garbage-in-garbage-out; there is little in database management systems that requires that the data be meaningful.)

Representing data as logical assertions, however, one is more inclined to regard these as statements about the environment. This leads to a consideration of the epistemological evidence behind these assertions, and the extrapolations and deductions that can be made from them.

A fundamental difference between logic programming and the more usual concept of theorem-proving is in the basic ontology. Theorem proving, following the usual pattern of logic, presumes some basic universe of discourse, e.g. numbers, blocks on a table. Logic programming, on the other hand, is much less restricted in this regard. In particular, much of logic programming is oriented towards objects that are data or syntactic structures. So, in addition to the more typical applications of predicate logic, logic programming may be used for example in sorting a list, or parsing natural language sentences. Thus, logic programming seems to blur the distinction between processing data structures and inferencing on logical assertions. For our purposes here, this ambiguity in logic programming serves as a useful bridge between the database management and logical views of databases.

G. RELATIONAL DATABASES AND LOGIC PROGRAMMING

Logic programming makes use of mechanical theorem-proving techniques as the basis for a general purpose programming language. The focus here is the use of logic programming for database inferencing.

The link between relational databases and logic programming is made by recognizing that, logically, a relation is the extension of a predicate. That is, a relation $P(x_1, ..., x_n)$ consists of all the n-tuples, $<x_1, ..., x_n>$, that satisfy the predicate, P. Thus, for example, the database:

EMPLOYEE	(NAME,	SEX,	SALARY)
	SMITH,	MALE,	35000
	JONES,	FEMALE,	42000

would be stated in a logic program as:

EMPLOYEE(SMITH, MALE, 35000).

EMPLOYEE(JONES, FEMALE, 42000).

Note that while the structure of the original relation is preserved, the attribute names are no longer used. Here the relation name, EMPLOYEE, is re-interpreted as a three place predicate and the attribute values in each tuple are its constant arguments. To refer to the entire relation, rather than individual tuples, attribute names might be translated as variables, e.g.,

EMPLOYEE(?NAME, ?SEX, ?SALARY).

However here, the former attribute names are merely arbitrary variable names. An equivalent designation would be:

EMPLOYEE(x, y, z).

Note how this example differs from its counterpart in the last section. Here EMPLOYEE is regarded as a single predicate, whereas before it was translated as a conjunct of three predicates. In conventional predicate logic the arguments of a predicate are normally regarded as names for individuals (in the universe of discourse). Here, on the other hand — and this is typical of most databases — not only do the arguments contain names for individuals, but other predicate names (e.g. MALE, FEMALE), as well as numbers (measurements).

As noted earlier, the ontology adopted by an organization (i.e., what basic individuals it recognizes) is a relative matter, depending on how it choses to view the world. (However, external reporting requirements may press it towards a more standardized ontology.)

In most databases, however, there is apparent recognition of an underlying ontology. This, again, is because data is generally retrieved in the same form that it is stored, without intermediate inferencing. For example, it is doubtful that any organization would regard FEMALE as naming a unique individual in the same sense that JONES does. But for database applications where logical inferencing is included, it becomes important to make this ontology explicit.

One of the simplest and perhaps most useful types of inferences for databases is for hierarchies of classification, so-called 'generalization hierarchies'. These were first proposed in the database management literature by Smith and Smith (1977), though they were discussed in Artificial Intelligence some years earlier, e.g. Quillian (1968). An alternative notation, based on the Entity-Relationship interpretation, is given in Lee and Gerritsen (1978).

A generalization hierarchy is a graphical representation of a sequence of subset relationships between categories. An example of the Smiths' (1977:109) is reproduced in Figure [2.1].

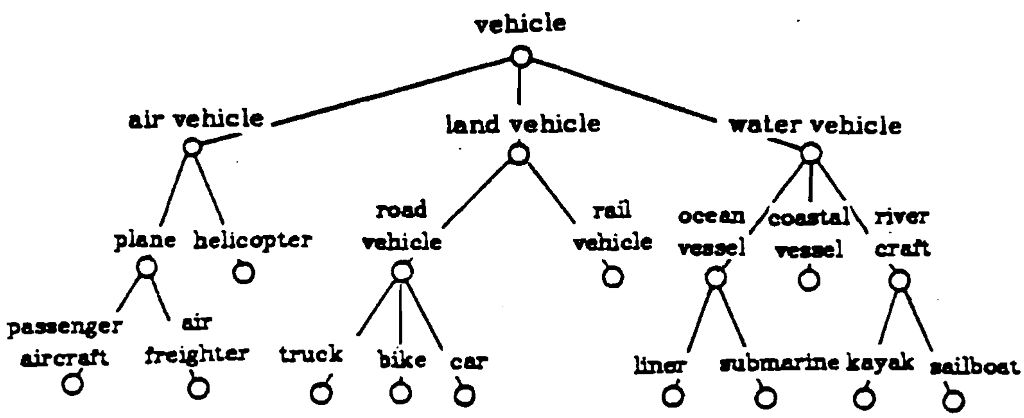


Figure [2.1]. A generic hierarchy over vehicles

The arcs in such generalization hierarchies are often read 'is a'. Thus an air vehicle 'is a' vehicle; a plane 'is a(n)' air vehicle; a passenger aircraft 'is a' plane; etc. Assuming that the primative predicates stored in the database are at the bottom of the tree, the generalization hierarchy translates into logic programming rules as follows:

PLANE(x) ← PASSENGER-AIRCRAFT(x).

PLANE(x) ← AIR-FREIGHTER(x).

AIR-VEHICLE(x) ← PLANE(x).

AIR-VEHICLE(x) ← HELICOPTER(x).

and so on. But now the ontological issues begin to emerge. Such inferences can only be made on relational attributes that are themselves predicates, and not, for instance, on attributes that are individual names or identifiers. In the terminology of the relational model, generalization hierarchies reflect the ambiguity that predicates may appear either as relation names or attribute values. These inferences are perfectly valid. However, they can only be recognized in the context of particular relations. Consider the following very simple example. Assume a relation:

EMPLOYEE(?ID, ?MARITAL-STATUS).

where ?ID is the employee's identification code and ?MARITAL-STATUS can have the values SINGLE, MARRIED or DIVORCED. (Note that the above expression is not a complete logic programming statement, as it is neither a fact or a rule. This expression could however be entered as a query, with a "?" following, which would then return all the tuples of the relation.) To plan office parties, we would like to specify:

ELIGIBLE(x) ← SINGLE(x) V DIVORCED(x).

However, having marital status as an argument of the employee relation, we are led to define it as follows:

EMPLOYEE2(?ID, ELIGIBLE) ← EMPLOYEE(?ID, SINGLE) V EMPLOYEE(?ID, DIVORCED).

The important thing to note is that we are forced to create a new relation name, EMPLOYEE2. The difference between EMPLOYEE and EMPLOYEE2 is that the latter has a different interpretation of its second argument.

The difficulty is that in typical relational form, with features (predicates) appearing as arguments, the governing predicate name carries the sense of these features implicitly. In the above example, the term EMPLOYEE carried not only the sense of employment, but also assertions about marital status.

Further deductive rules would entail the invention of further variants of the employee relation, e.g. EMPLOYEE3, EMPLOYEE4, each having its own peculiar interpretation of arguments. Hence, as the deductive rules become more complex, it becomes advantageous from the standpoint of conceptual clarity to promote these embedded features to the status of explicit predicates. Continuing the previous example, we would have:

EMPLOYEE(x) ← EMPLOYEE(x,y).

(Note: like-named predicates with different numbers of arguments are regarded as different predicates.)

SINGLE(x) ← EMPLOYEE(x, SINGLE).

MARRIED(x) ← EMPLOYEE(x, MARRIED).

DIVORCED(x) ← EMPLOYEE(x, DIVORCED).

ELIGIBLE(x) ← SINGLE(x) V DIVORCED(x).

Another type of data typically appearing in databases is numeric measurement. A similar rationale applies. As inferencing on the features indicated by these measurements becomes more complex, it becomes advantageous to separate out these features explicitly. For example, consider the relation:

BUILDING(?ADDRESS, ?HEIGHT-IN-METERS).

To convert to feet, we would like to specify the rule:

HEIGHT-IN-FEET(x,n) ← HEIGHT-IN-METERS(x,m), & n = m * 3.28.

However, as embedded in these relations, separate rules for the units conversion would be needed for each length attribute of each relation, e.g.,

BUILDING2(x,z) ← BUILDING(x,y) & z = y * 3.28.

Again we are faced with the introduction of the confusing terminology BUILDING2. Like before, the problem stems from the interpretation of the predicate name BUILDING to include more than the elementary concept of buildinghood, but also the measurement of that building's height. To distinguish these concepts explicitly, we would use the rules:

BUILDING(x) ← BUILDING(x,h).

HEIGHT-METERS(x,h) ← BUILDING(x,h).

(*) HEIGHT-FEET(x,z) ← HEIGHT-METERS(x,y), z = y * 3.28.

Having distinguished 'height' explicitly, we can now make use of this unit of measure conversion for other entities having the feature of height. For example, another relation might be:

PERSON(?ID, ?HEIGHT-IN-METERS)

To separate the concept 'person' from his or her height measurement, we add the rules:

PERSON(?ID) ← PERSON(?ID, ?H).

HEIGHT-METERS(x,y) ← BUILDING(x,y).

By using the rule (*) above, we may now infer the height in feet of any building recorded in the database.

Likewise, with the concepts 'building' and 'person' separately distinguished, we may want to add additional deductive rules about them. For instance,

PHYSICAL-OBJECT(x) ← PERSON(x) V BUILDING(x).

i.e. persons and buildings are both physical objects. With this abstraction, general knowledge pertaining to physical objects can then be added, e.g. that they have mass, height.

These examples reflect an important insight suggested by the graphical notation of generalization hierarchies. One would like to specify deductive rules to apply as generally as possible. For instance, it is a characteristic of vehicles of all types that they may change from one location to another. It is a characteristic of all water vehicles that their location will always be in some body of water. In normal database representations, one would have to specify these inferences repeatedly for each of 'submarine', 'kayak', 'sailboat', etc.

CHAPTER 3:

FORMAL SEMANTICS OF DATABASES

CONTENTS

A. DATABASE SEMANTICS

A key motivation in the growth of database technology has been the *integration* of information. For example, production and sales may both need access to inventory records. If they each keep separate copies, the two sets of records may become unsynchronized, resulting perhaps in foregone orders or frustrated customers. Consolidating the record keeping in an integrated database avoids this problem. Note however that this presumes that both sales and production have a common conception of what is meant by inventory. Normally this is not a problem since the two departments have had to interact long before the appearance of the computer, and so arrived (informally, naturally) at a common understanding.

This phenomenon is so ubiquitous that we seldom notice it until we change organizations. Then we may find that in the new environment, familiar phenomena are now designated by different terms, or that once familiar terminology now designates other things. Further, the translation is in many cases not straightforward, particularly in the language pertaining to the technical details of the enterprise.

As noted earlier, not only do organizations tend to differentiate themselves linguistically, but also this linguistic differentiation is an important component of their successful functioning.

However, opposing this tendency towards linguistic differentiation, there are also requirements for linguistic standardization between organizations. Contracts, for example, must be mutually understood; financial reports must be comparable to those in other organizations. The disciplines of law and financial accounting exist largely to provide this linguistic standardization between organizations. For good or ill, it is an empirical fact that governmental regulation is rapidly increasing in the US and Europe and elsewhere. Clearly the terminology of this regulation has to coincide with that used in the organizations being regulated. Efforts at international cooperation, reflected in international law, international trade agreements, international standards, etc., also depend on linguistic standardization. In these cases, multiple natural languages may also be involved.

The concern here is with those facts and communications that are likely to be channeled through an information system. The semantic problem can be viewed in terms of communicating from one information system to another, often called the problem of database translation. Given that the respective databases have been conceived in separate organizational environments, how can we ensure that data exchanged will be interpreted consistently in the two contexts? Kent presents a number of examples of this problem:

> A "book" may denote something bound together as one physical unit. Thus a single long novel may be printed in two physical parts. When we recognize the ambiguity, we sometimes try to avoid it by agreeing to use the term "volume" in a certain way, but we are not always consistent. Sometimes several "volumes" are bound into one physical "book". We now have as plausible perceptions: the *one* book written by an author, the *two* books in the library's title files (Vol. I and Vol II), an the *ten* books on the shelf of the library which has five copies of everything...
>
> IBM assigns "building numbers" to its buildings for the routing of internal mail, recording employee locations, and other purposes. One two-story building in Palo Alto, California, is "building 046," with the two stories distinguished by suffixes: 046-1 and 046-2. Right next door is another two-story building. The upper story is itself called "building 034," and the lower story is split into two parts called "building 032" and "building 047". IBM didn't invent the situation. The designations correspond to three different postal addresses: 1508, 1510, and 1512 Page Mill

Road are all in the same building. [1978:4]

A basic point, elaborated in later chapters, is that *this* problem, being semantic, is not subject to mechanical solution. Richer, more expressive database representation languages will not solve the problem, for it involves the relation of these languages to their referrents, not simply the languages alone.

It follows from the basic theory of automata and formal languages (e.g., Hopcroft and Ullman,. 1969) that the *most* an information system can do is syntactic transformations. The interpretations we apply to the symbols output from an information system rely heavily on their similarity to natural language vocabulary with which we are already familiar, either through our general education or through the narrower context of the organizational environment. Consider how much sense we could make from them if they were presented, say, using the Greek alphabet or as binary bit strings. How could we get the system to explain what was meant under these circumstances?

On the other hand, any conceivable problem of inter-connecting two information systems would take place in the context of a single natural language or at worst, two natural languages that were thoroughly inter-translatable. Thus, the translation problem should never really arise. However, it does. It does because in typical information system applications we are dealing at a level of semantic detail and precision that seldom arises in natural language discourse.

Consider a familiar topic like family relationships. We talk about families and relatives all the time with no apparent difficulty. Now consider the design of a census database. We want to count the number of families. We begin by developing a concept of a NUCLEAR-FAMILY. (The term is capitalized to indicate that we are developing a technical term to be used within the database). We begin with a notion that a NUCLEAR-FAMILY consists of parents and children living together in a common dwelling. Which parents? Multiple generations may be living together. Living together when? Suppose one parent travels frequently, or working abroad for a time or in the military service. Must the parents be married? What about co-habitating adults with or without children. (For this particular case, the US census bureau invented a special term, POSSLQ, pronounced 'possel-queue', abbreviating 'Persons of Opposite Sex Sharing Living Quarters'.

And then there is the problem of what counts as a dwelling. Must the dwelling be physically connected? Is it individuated by a single purchase or rental contract? What about families who have more than one residence? What about itinerant workers?

Suppose, to meet its purposes and interests, each national census bureau works out a suitable set of definitions for all the relevant concepts. Then some years or decades later the United Nations undertakes to combine these various databases to produce a world census. It would be an extraordinary coincidence if the various national databases were semantically compatible.

The census example is useful for the general familiarity of its subject. With regard to commercial organizations and other governmental agencies, the subject matter is often more specialized, but analogous problems arise when they must inter-communicate data.

It seems from this discussion that what we need to do in each of these cases of semantic incompatibility is arrange our terminology in a hierarchy of logical definitions, which reduce down, eventually, to some primitive terms that the parties involved agree on. An example of a logical definition might be:

BACHELOR(x) ::= MALE(x) & ~MARRIED(x).

(The symbol ::= is logically like the bi-conditional, ↔, but carries the extra-logical aspect of definition; i.e. the left-hand term is defined in terms of the right-hand expressions.) That is, a bachelor is defined as an unmarried male. This might be too simple. We might want to further include that a bachelor is a ***person*** who is male and unmarried; or we may need to specify marriage as a two-place predicate, etc. These are logical refinements that would depend on the purposes of the information system applications involved.

As long as what we are doing is ***defining*** the concept, BACHELOR, i.e., declaring it by fiat to be male and not married, this is perfectly adequate. It is presumably something along these lines that each national census bureau would do for its applications. This works fine as long as there is a single authority behind the definition making.

But the problem we have been elaborating is one where there has been more than one defining authority, and we are trying to reconcile the various definitions. For example, suppose that there are two databases. In the first, the term $BACHELOR_1$ appears and in the second, the terms $MALE_2$ and $MARRIED_2$ appear. The translation we are inclined to make is:

$$BACHELOR_1(x) \leftrightarrow MALE_2(x) \mathbin{\&} {\sim}MARRIED_2(x).$$

That is, wherever the left hand expression appears in the first database we translate that as the right hand conjunct in the second database. The problem is, *how do we know*? Supposing that these are primitive terms in the respective databases, further definitions won't help. But this is surely a case where the correspondence is indisputable. Winograd suggests some room for doubt:

> The word "bachelor" has been used in many discussions of semantics, since (save for obscure meanings involving aquatic mammals and medieval chivalry) it seems to have a formally tractable meaning which can be paraphrased "an adult human male who has never been married". Traditional theories of semantics deal with tasks such as determining whether the sentence "my bachelor uncle is unmarried" is analytic. In the realistic use of the word, there are many problems which are not as simply stated and formalized. Consider the following exchange:
>
> Host: I'm having a big party next weekend. Do you know any nice bachelors I could invite?
>
> Friend: Yes, I know this fellow X ...
>
> The problem is to decide, given the facts below, for which values of X the response would be a reasonable answer in light of the normal meaning of the word "bachelor". A simple test is to ask for which ones the host might fairly complain "You lied. You said X was a bachelor.":
>
> A. Arthur has been living happily with Alice for the last five years. They have a two year old daughter, and have never officially married.
>
> B. Bruce was going to be drafted, so he arranged with his friend Barbara to have a justice of the peace marry them so he would be exempt. They have never lived together. He dates a number of women, and plans to have the marriage annulled as soon as he finds someone he wants to marry.
>
> C. Charlie is 17 years old. He lives at home with his parents and is in high school.
>
> D. David is 17 years old. He left home at 13, started a small business, and is now a successful young entrepreneur leading a playboy's life style in this penthouse apartment.
>
> E. Eli and Edgar are homosexual lovers who have been living together for many years.

F. Faisal is allowed by the law of his native Abu Dhabi to have three wives. He currently has two and is interested in meeting another potential fiancee.

G. Father Gregory is the bishop of the Catholic cathedral at Groton upon Thames.

"Bachelor" was chosen here because it is the classic example of a logical (or analytic) definition. Winograd's remarks point out that even in this supposedly indisputable case, the sense depends on the interests and intentions of the parties involved. In cases of database translation, multiple organizations, hence multiple sets of interest, are typically involved. Each is likely to have attached its own specialized sense to its terms, relative to those interests and each organization's own internally developed technology. Of course not all the terminology in each organization's vocabulary is candidate for database translation. An important research issue is to determine in which areas inter-translatability is most needed, and to focus on the semantic foundations in these cases. (The comments in chapter [6], on the semantics of accounting, apply to this.)

B. PROBLEMS OF SEMANTIC CHANGE

The database translation problem is the one typically cited to motivate semantic issues. This reflects the underlying operational orientation of database management, which concentrates largely on production and/or sales related transactions. However, a deeper and more important problem exists, namely semantic change within the organization itself.

It is a commonplace to observe that the world is changing rapidly. Organizations, to survive, must keep pace, and to succeed, must innovate. This entails not just a re-combination of old concepts, but changes in the concepts themselves. Thus the concept of automobile changes year by year as new models come out. The term 'computer' originally referred to a person who computes (which is why ACM abbreviates Association of Computing Machinery, to avoid confusion with human computers). Then 'computer' came to mean a big machine filled with vacuum tubes. Now we think of Apples and wristwatches. Television ads constantly press us to consider new conceptions of soapsuds, breakfast food, toothpaste; fashion changes our conception of apparel; etc. Managers participate in these changes in their understanding of the markets, changing technology, social trends, politics, etc. Given that management behavior is almost entirely linguistic, conceptual change involves semantic change.

However, computational inference generally entails an assumption of stable semantics. For instance, a logic programming rule,

$$Q(x) \leftarrow P(x)$$

is valid or invalid depending on the semantic extension of P and Q. For example, if P is lemon and Q is fruit, the conclusion is correct, since anything that is a lemon is also a fruit. If P is interpreted as 'elephant', however, the rule is invalid. The problem created by semantic change is that the inferences made by the system, once correct, become invalid as the extension of the symbol changes. For instance, the rule

$$PERSON(x) \leftarrow COMPUTER(x)$$

was once valid but is no longer. This has deep consequences for the use of information systems in organizations in dynamic environments. As will be developed in later chapters, the problem is closely associated with the more general issue of rationalization in organizations.

C. USE VS MENTION

Database management arose from data processing concerns about storing large amounts of data on magnetic devices. As such, the attention was on the data itself as an object, rather on than what it meant.

It is interesting to note how the usage of the very term "data" has changed. "Data" was originally the plural of "datum", hence taking a plural verb ("the data are ..."). Now we use "data" more as a mass noun, ("the data is ...") suggesting that we have come to view data more like a fluid that flows, is stored, processed, etc.

To view data as objects, or perhaps even in fluid terms, is key to the engineering of the data processing. As a highly automated industry, data processing shares many of the efficiency problems of other process industries, e.g., oil refining, food processing.

But here we tend to lose sight of our basic product, data as *facts.* Linguists refer to this as a confusion between *use* and *mention*, e.g.,

snow is white

vs

"snow" is a four letter word.

In mentioning a term, as in the second case, we are concerned only with its form, not in its use as a symbol for something else.

The position taken here is that the role of information technology in organizations can be better understood if we avoid discussion of 'data' entirely and focus rather on the content of databases as assertions in a language.

Another aspect of databases that tends to confuse these semantic aspects is that they are intimately tied to the programs that process the data or, in the terminology here, that provide inferences on the elementary assertions in the database.

With respect to programming languages (i.e. L_C in chapter [1]) there is a well developed concept of semantics, but it is quite different from the one sought here.

This merits a brief explanation. programming languages are used to issue commands, hence imperative statements, to the computer. The semantics of an arbitrary imperative like,

Shut the door!

depend on understanding what objects are involved (e.g., door) and the change of state requested (the door is open, then the door is closed). In programming languages, the objects involved are data structures, ultimately memory locations, and the changes of state are the bit status in these locations. Thus the semantics of a programming language depends only on these computational objects.

To discuss the semantics of these data structures in the organizational environment, we have to regard them as symbolic expressions, representing objects in the external environment, rather than objects in themselves, returning once again to the use/mention distinction.

D. FORMAL LANGUAGES

In the last chapter, relational databases were re-formulated as first order logical assertions. This had on the one hand the advantage of linking databases with logical theorem proving apparatus, enabling database inferencing. On the other hand, this reformulation also presents databases in a perspective that highlights their semantic aspects in a way that can benefit from a large background literature on semantics as it relates to logical languages.

Whereas database management is only now beginning to clarify its semantic problems, many of these have already arisen in the study of logic and formal linguistics. A fundamental concept is that of *formal language*. A formal language is one who's use is controlled by explicit rules. This contrasts with natural languages (e.g., English) who's use is a matter of evolving, implicit consensus. The aspects of a formal language are typically divided into the following categories:

a. the *syntax* of the language, consisting of

   i. a *vocabulary* of basic symbols (i.e., its words and punctuation)

   ii. *formation rules*, which determine permissible ('well-formed') combinations of the vocabulary.

   iii. *transformation* or *inference rules*: these provide a mapping between true sentences in the language; i.e., they are truth preserving transformations of the form: *if* $A_1, \ldots, A_n$ are true sentences, then $B_1, \ldots, B_m$ will also be true sentences.

b. *semantic rules*: these are rules mapping expressions in the language to the objects they denote. These rules follow the so-called 'Principle of Compositionality': the semantic rules determine the denotation of more complex expressions as a composition of their syntactic components, based ultimately on the denotation of the basic vocabulary. The structure of the semantic rules therefore follows that of the syntactic rules.

E. PROOF THEORY VS MODEL THEORY

Of special interest is how formal languages are used to reason about the world. Proof theory and model theory present dual explanations of this process. These might be characterized respectively as syntactic vs. semantic reasoning.

Proof theory is the mode of reasoning generally associated with logic. It is concerned with deductions based solely on the syntactic form of expressions and, in particular, does *not* make use of any semantic information.

Model theory, by contrast, is concerned with reasoning in the formal language based on the denotations of their expressions. For instance if, by observation, we discover that the class of boys is a subset of the class of males, we might infer in the formal language that if BOY(x) then MALE(x).

In terms of the three components of formal languages just discussed, the duality between proof theory and model theory might be diagrammed as in Figure [3.1].

A *theory* in a formal language is a collection of statements asserted to be true about some discipline or subject area. Importantly, databases constitute a theory of a (first order) formal language. Tarski, the principle developer of this dual view of proofs and models, sets forth the basic strategy of proof theory as follows:

> ... if within logic or mathematics we establish one statement on the basis of others, we refer to this process as a *derivation* or *deduction*, and the statement established in this way is said to be *derived* or *deduced* from the other statements or to be their *consequence*.

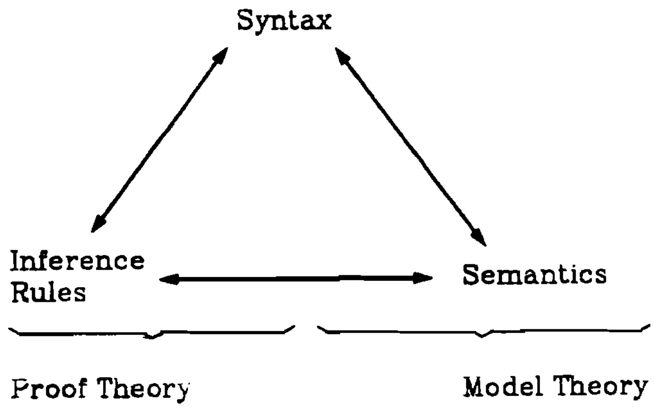


Figure [*3.1*].

> ... The method of constructing a discipline in strict accordance with the principles laid down above is known as the ***deductive method*** and the disciplines constructed in this manner are called ***deductive theories***. [Tarski, 1941/1969:118-119].

The problem with proof theory is that it is rather dogmatic about the concept of truth. The contention is that *if* the axioms are true, ***then*** the theorems deduced from them will also be true. However, in proof theory by itself there is no independent way of verifying this claim. Model theory is an effort to provide this independent verification, particularly in the case of first order languages. As it turns out, the method of truth tables is a special case of model theory for verifying the semantics conditions of propositional (zero order) languages. Here, the denotation of a proposition (sentence) is taken to be a truth value (***true***, ***false***). The concept of a tautology, for instance, could be demonstrated as a compound sentence that remained true under all possible assignments of truth values to its components. This illustrates the basic concept of a ***model*** of a formal language: it is one unique interpretation of the language that makes an assignment of denotations to its basic vocabulary.

In the case of first order languages, an assignment of denotations is more complex since the basic vocabulary consists not just of whole propositions but rather of individual constants and predicates. The view here is that these denotations should all be definable in terms of a set of individual objects, called the ***domain of individuals***, D.

a. Individual constants in the language denote elements of D.

b. One place predicates denote subsets of D.

c. n-place predicates denote n-ary relations defined on D.

Clearly, the denotation of individual constants depends on the choice of D. However, once D is defined, the denotation of predicates further depends on the way one decides to define subsets and relations on D. Hence, a model, M, of a first order language has two components: first, the choice of a set D; but also an assignment function, F, which maps individual constants to elements of D, and predicates to subsets and relations on D. A small example may help. Consider a mini-language, L, with the following basic vocabulary,

Individual constants: A, B, C, D.

Predicates: 1 place: P, Q

2 place: R

Let D = {John, Mary, Ted, Alice}

One assignment might be:

den(A) = John

den(B) = Ted

den(C) = Mary

den(D) = Alice

den(P) = {John, Ted}

den(Q) = {Mary, Alice}

den(R) = {<John, Mary>, <Ted, Alice>}

'.LP

Consider that we interpret P as 'male,' Q as 'female,' and R as 'married.' Suppose that it is the case that John is married to Mary and Ted is married to Alice. Then in this particular assignment, the assertion

R(A,C).

is true.* However, clearly we could have made other assignments of the basic vocabulary where this would not be true.

The import of this for logic is that it provides a clear and exacting distinction between contingent vs logical truth. A contingent truth is one that holds for a particular choice of D and F — what Tarski termed ***true in a model***. A logical truth (the analogue to a tautology in the propositional case) is one that is true for all choices of D and F, i.e., in all possible models. What model theory does, basically, is define the semantics of first order languages in terms of sets. Further enrichment of model theory continues with the interplay between the syntactic complexity of

* The reader may be confused between the notation of predicates sometimes with arguments, sometimes without. An intermediating logical device is provided by Church's lambda operator, where, for example, P abbreviates $(\lambda x)\ P(x)$, R abbreviates $(\lambda x)(\lambda y)\ R(x,y)$. Lambda abstractions denote the sets for which the predicate is true.

the language and its corresponding set theoretical mathematics.

F. MODELS OF DATABASES

As observed earlier, a database can be construed as a theory (collection of sentences) in a first order language. Coupled with the earlier remarks about theorem proving on databases, the relevance of model theory emerges as a means of semantically verifying database inferences. This gives a tantalizing insight to that old DP adage: Garbage-In-Garbage-Out. In the terminology just presented, computers can provide proof theoretic deduction but lack a model theoretic verification. (Even with the addition of robotic sensors this situation may not change very much — see next chapter.)

However, if one examines the literature on model theory since Tarski (e.g., Chang and Keisler, 1973), one finds that it is almost entirely mathematical, that is, the sets that form the denotations of various formal languages are characterized as discrete, continuous, finite, infinite, etc., but little more is said about their 'real world' character. In keeping with this mathematical orientation, the composition of these sets is presumed to be entirely known when the language semantics are defined.

In database contexts, this would correspond to the situation of a static or 'snapshot database', e.g. a census database where once the data is collected it is no longer updated. That is, the population of individual objects, their properties and relationships are entirely fixed. In these cases, the denotation of terms like ADULT or HEAD-OF-HOUSEHOLD is determined by the population of the known universe or nation at the time the database 'snapshot' was taken.

However, the difficult aspects about databases arise from the fact that they are more often dynamic, undergoing continual updates. The denotation of a predicate like EMPLOYEE changes from week to week. We cannot define the semantics to be for example the set of employees at a given time, or even the set of employees during the entire history of the firm, for that still would not include the very important class of future employees.

This presents important problems if we are seeking a semantic justification for some generalization like 'all employees are people'; 'all salaried employees have health insurance'; 'all executives drive automobiles'. The point is, we would like to make these generalizations *now*, but be able to draw inferences about health insurance, etc., *later*. That is, we want our generalizations to apply to the database at present, its past states, but also its future states. Stated slightly differently, we want to generalize not only about what *is* true, but about what *can* be true or what *must* be true. The technical term for these are the so-called *alethic modalities* of possibility (can) and necessity (must). These are commonly regarded as dual concepts:

necessarily p	↔	not possibly not p
possibly p	↔	not necessarily not p
impossibly p	↔	necessarily not p
not necessarily p	↔	possibly not p

But now compare the statement

all ravens are black

to the statement:

necessarily, all ravens are black.

What does the word 'necessarily' add? The generally accepted response to this was given by Kripke (1963) that an assertion 'p' is a claim about the actual world, i.e., within empirical experience, whereas 'necessarily p' is an assertion about all possible worlds.

Whether an assertion is true in all possible worlds depends both on our current scientific theories about the world as well as the definitions we assign to terms. This point will be elaborated at more length in the next chapter. For the moment consider the concept of a possible world to be like a gedanken experiment — i.e., it is a world (or state of the world) in our imagination. Hence, 'necessarily, ravens are black' is true if the set of ravens is a subset of the set of black things in all the world states we can conjure up. However, if we can imagine, for instance, an albino raven in one of these states, then we would deny ravens are necessarily black.

The concepts of necessity and possible worlds interact with that of time. We can imagine alternative versions of the present world, alternative histories, as well as multiple scenarios of the future. In databases, however, we are in the main only concerned with the actual past and present. However, fortune telling and economic forecasting both being somewhat dubious, we have no corresponding concept of the actual future.

The inter-relationship between possible worlds and times can be pictured graphically as follows. Assume an elementary vocabulary of propositions P_1, P_2, ..., P_n, which describe all relevant aspects of the world. We assume these to be logically independent, that is, the truth value of one does not affect the truth value of any other. Then we can invent the concept of a state of the world as a conjunct of these propositions (von Wright, 1968). By assigning different truth values, we thus generate 2^n possible states.

Presuming that different states can be true at different times, we arrive at a state/time grid as in Figure [3.2]. The actual world will presumably be reflected as one sequence of states, i.e. the solid line on the grid. Other possible worlds will be the other paths that can be drawn connecting states progressively through time, e.g. the dashed line in Figure [3.2].

In the business world, where the past is regarded as a sunk cost, alternative histories are of little interest. Thus, for example if we take T_5 to be the present time, what is of interest are those possible worlds whose history conforms to the actual world, but diverge in different directions towards the future (see Figure [3.3]). By re-drawing the graph so that possible worlds, rather than states, are indicated by the horizontal lines, this conception of time and possible worlds is characterized as 'backwards linear and forward branching' (Rescher and Urquhart, 1971, ch. 8), see Figure [3.4]. The relationship between time and possible worlds just described is over-simplified in that it assumes the elementary vocabulary to be whole propositions rather than predicates applying to individuals. Thus in generating possible states, we would need not only to make assignments of truth values, but also assignments of individuals to logical names, to predicates, and so on. This is exactly the definition of a Tarskian *model*. That is, a model is a possible state of affairs. A possible world, as we have defined it, is a sequence of such states over time.

This raises one nit-picking little point that ends up being the cause of enormous amounts of philosophical debate. The Tarskian concept of a model assumes we have some domain of individuals, D, which are the extension of names, predicates, etc. We describe states of the world as different arrangements of properties on these individuals. But how do we know they're the same individuals from one state to the next if they have different properties? Are there certain properties that are essential to the identification of the individual while others are accidental, varying over time? If so, which ones? This is known in the literature as the problem of 'trans-world identification of individuals', discussed in more detail in chapter [4].

G. DATABASE ONTOLOGIES

The question of the identity of individuals through different states of affairs presumes that we know what an individual is in the first place. This is the issue of *ontology*. To elaborate this in terms of databases, recall that Chen, in his Entity Relationship interpretation of the Relational Model (see Chapter 2), proposes that database relations are essentially of two types —those that apply to single entities, e.g.

EMPLOYEE(?E-NUM, ...)

DEPARTMENT (?DEPT-NUM, ...)

and those that assert a relationship between entities, e.g.

WORK-FOR(?E-NUM, ?DEPT-NUM)

(?E-NUM and ?DEPT-NUM are unique identifiers for employees and departments, respectively.) In reformulating this in terms of a first order language, this distinction translates to the fairly straightforward distinction between single and multiple place predicates. But the question then arises, 'what is an entity'? People are entities, but are colors? are numbers? Recast as a first order language, the issue focuses on the composition of the domain of individuals, D. The choice of D is called the ***ontology*** of the theory being formalized. It represents the key foundational assumption of the theory. (See Quine 1953/1961.) For instance, one theory might assume D to be the set of positive integers. Another might take D to be the set of discrete physical objects. Thus the ontological question, from the standpoint of formal languages, is moot. It is an assumption of the theory, thus taken for granted. If we were considering databases only in isolation, this would be satisfactory. The database designer decides what to regard as entities, and the semantics of the database follows from that choice.

However, to make use of these data resources in decision support applications involves their combination with analytical routines, often in ad hoc ways. We would like the system to know which things can, semantically, go with which. But this requires that they have a compatible ontology. If, for example, the ontology of one system recognizes consumer tastes as the only elementary entity while another recognizes only physical objects (e.g. parts for furniture), it will be difficult to reconcile the two for, say, aiding marketing decisions.

One ontological issue can be fairly easily dismissed at this point. This relates to the use/mention distinction introduced earlier. Most data models regard their ontology to consist of symbolic constructs, e.g., character strings and (real, integer) numbers. This is appropriate for software research, where the attention is confined to the information system itself. It does not serve, however, for applications of the technology to the organization's problems. Here character strings and numbers are parts of the language used to describe the organization and are not in themselves of interest.

Thus, in the discussion of logic programming in the last chapter, numbers were introduced as a special type of individual. This is more of a syntactic convenience than an ontological issue. For instance, the two place predicate, AGE(x, n), could just as well have been regarded as a family of predicates, e.g.

AGE30(x)

AGE31(x)

AGE32(x)

etc. A similar case can be made for character strings. For example, in most PROLOG implementations character strings are designated as an arbitrary sequence of characters between double quotes, e.g.,

"THIS IS A CHARACTER STRING."

These might be used in database applications to accommodate alternative (and not necessarily unique) identifiers or labels for individuals. For example, employees typically have an employee identification code assigned to them by the organization. They might also have a social security number, a driver's license number, etc. The employee's first and last names are non-unique identifiers. Thus, to record these various identifying references, we might introduce character strings as a different type of individual, serving to distinguish the labels used externally from the internal logical name. For example,

SSN(JOHN,"521-37-5126").

LAST-NAME(JOHN, "SMITH").

indicate that for the individual known internally as JOHN, external labels for this individual are his social security number, 521-37-5126, and last name, SMITH. Again, this is more a syntactic convenience than an ontological issue. The role of character strings in this case is simply to mediate between internal and external naming schemes.

H. INTENSIONAL ENTITIES

There is one other body of work that should be mentioned in this connection — so-called 'Montague semantics', after the original work of Richard Montague (see for example, Thomason, 1981, Dowty, et. al., 1981). This work has become of great interest in linguistics as promising a mathematically tractable theory of natural language semantics (thus complementary to the formal syntactic theories of Chomsky, etc.). Montague, a one time student of Tarski, attempts to provide a model theoretic explanation of reasoning in natural language discourse.

Of particular interest to Montague are inferences relating to so-called intensional contexts* including aspects of belief, expectation, intention, etc. For instance, from "John believes the world is flat" we do not infer that the world is flat. More subtle cases also arise, for example "the temperature is 90 and rising" does not entail that the number 90 is rising.

These problems often seem rather esoteric to non-linguists. One aspect however has bearing on the issue of database ontologies. This is Montague's conception of intensional entities. It often happens in ordinary language that we speak of properties, such as red, as if they were entities in their own right, e.g., 'Red is my favorite color.'

This cannot be waved aside as a matter of ontological choice since people tend to mix these references in the same discourse — e.g., when shopping for a dress or a shirt one regards color as a property of these objects, but also expresses preferences for colors independently of the

* note the two spellings: 'intension' vs 'intention'.

objects that have these colors.

The essence of a property, e.g., redness, has historically been termed its *intension*, as opposed to the *extension* of the property, which is the set of objects for which the property holds. (The historical development of this distinction is due mainly to Carnap; see for example Hintakka, 1975.) Montague, in observing how we apparently reify properties to the status of entities, wanted a mechanism to make these intensional aspects extensional.

Earlier, (in a footnote) we mentioned Church's lambda operator as a device that mapped a predicate to the set of entities in the domain D that it satisfies. For example,

$$(\lambda x)\ \text{RED}(x)$$

denotes the subset of red objects in D. This however refers only to the current population of red objects and is insufficient to explain 'redness' in the abstract. Montague's claim was that the intension of a property is its extension not only in the actual world but in all possible worlds at all times.

To express this he proposed the intension operator, $^{\wedge}$. The effect of this (see Dowty, et. al., 1981, ch. 6) is to repeat the lambda abstraction not just on the domain D, but also across the domain of times and the domain of possible worlds. Hence, the expression

$$^{\wedge}\text{RED}$$

denotes the set of red objects in all possible worlds and times.

This helps to clarify our thinking about database ontologies. Even with a basic ontology of, say, physical objects, properties such as red can be regarded as entities by this semantic device. Indeed, this is the sort of thing that seems to occur in many scientific discussions. For example, the farmer says, 'the cat is lazy,' referring to his particular cat. The biologist says, 'the cat is warm-blooded', referring to an intensional concept of cat.

On the other hand, while the preceding explications of necessity and intensions offer a pristine, mathematical elegance, we are left with a certain discomfort that reference to infinite sets of possible worlds just will not sell very well in the earthy, mundane world of management. Much of the problem, it seems, has been waved away in the facile assumption of possible worlds. Bringing these back down to the ordinary reality where we actually live is the subject of the next chapter.

CHAPTER 4:

NAMING: INDIVIDUALS AND NATURAL KINDS

CONTENTS

A. REFERENCE

It is not only with respect to databases that one can find dissatisfaction with model theoretic semantics. There is also a movement within analytic philosophy to bring these semantic theories more into accord with the mundane mechanisms of language use. This movement focuses particularly on aspects of *reference*, how a symbol is linked to its denotation. This pertains both to the naming of individuals and the naming of classes.

Of particular interest here are the categories we use for our everyday objects, so-called *natural kinds*, such as lemon, bottle, chair. When examined through the perspective of model theoretic semantics described in the last chapter, the existence of such kinds is puzzling. This puzzlement arises largely as a result of adopting set theory as the principle device for explaining how real world objects are organized.

The problem is essentially that any collection of objects can constitute a set — e.g., the set consisting of my toothbrush, the Eiffell Tower, and the planet Saturn. Given all the possible sets of things, why are some, e.g., lemons, chairs, given a special status and assigned a name? The problem becomes all the more complicated if the dimensions of time and possible worlds are added. (E.g., an arbitrary set might then include hypothetical individuals such as Abraham Lincoln's automobile, the present King of France and the city of Atlantis).

An early reply to this problem was that natural kinds were sets defined intensionally. That is, there were certain 'critical properties' that selected the members of these sets. An obvious problem with this view is that, from the standpoint of model theoretic semantics, this involves a circular argument: One cannot explain properties by their set denotations and then turn around and explain the sets denoted by their intensional properties. However, this merely casts doubt on the denotational approach to semantics and suggests that perhaps intensions should be taken as primary after all.

However, the criterial properties approach also quickly runs into difficulties. Consider the concept of a chair. What are its criterial properties?

a. that it has four legs? No, there are chairs without four legs.

b. that it has a horizontal surface and a vertical back? No, for instance a bean bag chair has neither of these.

c. that it is something to sit on? We can sit on many things that aren't chairs.

This view is conjunctive — it requires that each element of the intended set satisfy the several criterial properties simultaneously.

An alternative view is disjunctive: that there are no single properties that run throughout the set of things we call chairs, but it is rather the disjunction of several properties that define this set. For example, a chair is:

four-legged *or*

has vertical and horizontal surfaces, *or*

is used to sit on, etc.

The problem here is that it tends to include too many things —e.g., tables would count as chairs by this definition. Wittgenstein (1953/1958) is a classic philosophical discussion of the shortcomings of the criterial properties (or essentialist) view of semantics.

More recently, attempts have been made to get around this problem by saying that the denotation of natural kind types of predicates is a fuzzy set. Without debating the adequacy of fuzzy set theory, we observe merely that this misses the basic point. The problem is not whether the boundaries of these sets are sharp or fuzzy, but rather why they are selected and named in the first place.

Providing more pragmatic motivations for these sorts of semantic issues, Kent (1978) cites numerous examples arising in data processing applications. For example, consider the natural kind, street:

> What is one street? Sometimes the name changes; that is, different segments along the same straight path have different names. Based on a comparison of addresses, we would probably surmise that people on those various segments lived on different streets. On the other hand, different streets in the same town may have the same name. Now what does an address comparison imply?
>
> Sometimes a street is made up of discontinuous segments, perhaps because intervening sections just haven't been built yet. They may not even be on a straight line, because the ultimate street on somebody's master plan curves and wiggles all around. And sometimes I can make a right turn, then after some distance make a left turn an be back on a street with the same name as the first. Is that one street with a jog? When do we start thinking of these as different streets having the same name?

Problems of this sort have recently come to focus in the works of such philosophers as Kripke (1971, 1972) and Putnam (1970, 1978). Schwartz (1977), a collected edition on this subject, dubs it 'the new theory of reference'.

The discussion in this chapter will therefore focus initially on individuals and the epistemic aspects of proper names. Building on that, the recognition and naming of natural kinds is considered. These depend on social conventions that, particularly for economic goods, change over time. Social movements and economic innovation are reflected in linguistic changes. Organizational adaptation likewise depends on linguistic evolvability. Structured information systems, relying on a fixed, stable semantics, constrain this evolution.

B. INDIVIDUATION

In the last chapter, ontology was discussed as the choice of the domain of individuals, D. The elements of this set are called simply individuals. In database management, the term 'entity' is more frequently used, but this has the drawback that it tends toward a certain confusion between a particular entity or the generic class (e.g., EMPLOYEE as an entity).

Understanding natural kinds, it turns out, depends on understanding the conventions for recognizing and naming the individual entities included in the kind. The recognition of a single individual is called ***individuation***. This has a static and dynamic aspect: recognizing the individual at a point in time, and recognizing that individual as it undergoes change. The importance of individuation is that it is the criteria by which we assign names or identifiers to an individual. To use a name (in a formal language or a database) presupposes that all of the users of the language/database agree on the object designated by that name. Thus, the problem of individuation has its dual in the problem of sameness. If a and b are names, a = b asserts that these name the same object*.

A thorough study of the various aspects of individuation is Strawson (1959). He claims that our basic criteria for individuation is the object's location in a spatial/temporal framework. Hence, the objects easiest to individuate with reasonable consensus are e.g., physical objects and events somehow related to physical objects. This is a very important observation since it gives us some insight into areas where individuation is likely to be difficult, namely abstract objects not involving space/time locatability.

Consider for example Beethoven's 5th Symphony. What is the designation of this term? Is it the event of Beethoven's composing this piece? Is it the paper it was originally written on? Is it the collection of all paper reproductions, etc. of this original? Is it a musical performance of this piece? Is it all musical performances, past and future of this piece? We would like to say that it is none of these. Beethoven's 5th Symphony is an idea, and these examples are all mere conveyances of this idea.

A more modern example, coming into increasing economic importance is the individuation of computer programs. Like the symphony, computer programs have static representations on disk, in core, etc. as well as performances — in the execution of the program. Yet, again we would like to claim that the computer program is actually an idea.

* Two sameness problems are sometimes distinguished: sameness of individuals vs sameness of kind. For example, "John and Bill drive the same car" may mean they drive the same particular car, or that they each drive distinct cars that are of the same type (satisfy the same predicates). Insofar as the extension of a predicate is a set of individuals, sameness of kind also relies on individuation.

Consider the problem of software theft. Typically when something is stolen, the owner suffers its loss. In the case of stolen software, however, the owner often can't even detect the loss. Theft is probably the wrong word here — it is actually more like plagiarism. However, the point is the same: if we want to talk about particular symphonies or computer programs, it is very hard to do so without relating them somehow to a spatio-temporal framework.

C. SPATIAL EXTENT OF INDIVIDUALS

The world, says Quine (1964:4), consists of middle-sized objects. This is certainly true of the individuals typically identified in databases. One aspect of this problem relates to parts decompositions. For instance, a car consists of a body, motor, tires, etc. The motor in turn consists of engine block, crankshaft, pistons, carburetor, etc. The carburetor in turn consists of valves, etc. In this example, each of the parts is detachable and replaceable in the whole. This is generally the basis for our interest in parts in the context of database applications (though one may equally well talk about non-detachable parts — e.g. the front of a house).

If sets are used as the basic construct for organizing reality in these cases, one encounters what seem to be unnecessary complications — e.g., regarding a car as a third order set. A useful approach to this problem is provided by Goodman (1951/1977) in his *calculus of individuals* (sometimes called ***part/whole*** theory). The basic idea here is to discard the set theoretic distinction between 'element-of' and 'subset-of' and replace them by the single predicate, 'part-of.' In this view, collections of individuals simply constitute another first order individual. This concept of individual, to Goodman, is intended to be equally general as the concept of a set. For example, the collection of all people, past and future, can be regarded as a single individual.

It is debatable whether part/whole theory has advantages over set theory generally (e.g. as an alternative basis for mathematics). However in certain contexts, like those just mentioned, it seems to correspond much more closely to our intuitions.

Another problem with spatial extent involves tiny objects that are too numerous or too cheap to bother naming individually. Examples are grains of all types, nails, light bulbs, etc. Here, the individuating device is typically a *container* — e.g. a box, inventory bin, whose collective contents are big and/or valuable enough to name individually. Similar comments apply to liquids.

D. TEMPORAL EXTENT OF INDIVIDUALS

Even when an object can be individuated spatially, its individuation across time is sometimes problematic. A classic example in philosophy is the so-called boat of Theseus. Imagine a wooden boat. We replace a plank of the boat, setting the old plank aside. Is it the same boat? We replace other planks, one by one, until all the planks have been replaced. Is it still the same boat? If not, which replacement caused it to be a different boat? But now, we take the planks we have set aside and build a new boat. Is this not the original boat since it is composed of all and only those parts in the original? (To aggravate the argument, we can iterate this process to create an entire navy of apparently identical boats.)

Many other examples can be found from database applications, e.g. replacement of parts, phases of manufacture, remodeling of buildings, reorganizations, mergers of corporations. In many of these cases, there seems to be no essential criterion that determines the temporal extent of the entity*.

Fortunately, the problem for databases does not depend on an absolute answer, only on a consensual one. The important issue is the correspondence between the name we have for the individual as it undergoes transformations throughout time. In particular, if we have an individual which today we name X, how will we recognize that individual tomorrow?

In the case of (middle-sized) physical objects, the problem is often resolved by imprinting or tagging the individual with an identification number, e.g., serial numbers on vehicles, inventory codes on office equipment, room numbers, street addresses or buildings.

In the case of persons, the continuity of naming is generally maintained in the person's own memory. For example, a baby is named 'John Doe' by his parents. In his baby years he learns that name as his designator. As an adult, when I first meet him, he tells me, "My name is John Doe". When he phones me a month later he says, "Hello, this is John Doe", and so on.

* Note: living things seem to be the one major exception here. We take the ongoing process of life to mark the continuity of the individual, even though, like the boat, all of its cells may eventually be replaced. However, even here difficulties are beginning to arise, particularly surrounding the morality of abortion. Does the human individual begin at conception or at birth?

People can change their names. They typically provide the continuity of identification by telling you — e.g., "My name is Mary Doe — I used to be Mary Adams before I was married".

There are, generally speaking, social incentives for a person to consistently report his name through time. It is principally through this convention that the person is known to the various social institutions. Also, because of the dependence on this convention, criminals can sometimes 'change their identity' by altering their physical appearance, but most importantly, by reporting a different name when asked.

For data processing efficiency, and to avoid the problems where ordinary names are accidentally duplicated, organizations often provide their employees, clients, etc. with identifying codes — e.g., an employee number, social security number. Normally, these identifying codes are connected to the individual in a way similar to ordinary names, i.e., the person consistently reports the same code.

In other cases, where there might be incentives for mis-reporting, a further device is required to perpetuate the association of the code with the individual. A common example currently is credit cards. Here the perpetuating device is the physical possession of the card.

A macabre example from the past was in concentration camps. Prisoners were physically tattooed with their identification number. Not only did this prevent them from misrepresenting their identity, they could also be identified after death.

In the philosophical literature, these social mechanisms by which a name continues to be associated with the thing it designates are called 'causal chains.' These have been used to explain philosophical puzzles in such sentences as:

'Mark Twain is Samuel Clemens'

'The Morning Star is the Evening Star.'

Normally if two names uniquely designate a common object, they should be interchangeable in any context. If that were true, the above sentences would be tautologies. The fact that they indeed convey information is a result of having different causal chains associating each name with its referent.

Our acquaintance with a particular individual is typically not continuous across time. We see a friend one day, again a week later, etc. Further, the sense data we have of that person is often incomplete and highly varied —we see the person with different dress, in different lights, different angles, distances, etc. That is, our sense data of that person amounts to samplings of different aspects of his/her physical appearance, voice, manners, etc. However, we need very little of this data to infer the continuity of that person through time. Lacking anything else, the consistent reporting of a name is often sufficient evidence for us especially if 'we' are an organization or institution. A college story goes that a fraternity enrolled their mascot dog in the university, putting his name on exams, etc., until he was finally graduated with a bachelor's degree. The story may be false, but it makes the point: an institution's acquaintance to individuals is based heavily on the reporting of names.

E. NATURAL KINDS

The preceding explication of identity assertions perhaps does not seem very surprising to anyone experienced in data processing. For example, the assertion,

CUSTOMER #12 = SUPPLIER #57

connects the causal chains between the way we learn about customers and the way we learn about suppliers.

What may seem surprising is that this same mechanism of causal chains is used to explain the naming not only of individuals, but also of natural kinds. Earlier in the discussion of Goodman's calculus of individuals, we observed that a set of objects could alternatively be viewed as another (collective) individual. This should not seem very surprising — e.g., we often regard football teams, departments, or forests as individuals composed of other individuals.

Now consider the case with water. Normally we do not deal with water individuals at the molecular level, but rather with water individuated in collective units or containers, e.g., water droplets, a cup of water, a puddle of water, a lake, an ocean. Mentally, we can easily conceive of emptying smaller containers into larger ones to form a larger water individual. Now if we consider that most of the oceans and seas are interconnected, we can fairly easily come to imagine the world as a very large water container. Further, our concept of the water it contains is a fairly permanent one. We think of the water in the world as going through various transformations (snow, ice, vapor, steam), but its sum total on the planet is basically fixed through time (ignoring molecular transformations). Is there then a sharp difference between a water individual and the natural kind, water? One might object that the kind, water, would also include water on other planets, etc. However, that only requires us to imagine a larger water container.

At the close of the last section, it was noted that we come to know the features of an individual (person, etc.) through a series of occasional glimpses, each conveying certain aspects of that individual. However, where our own sense impressions are not sufficient for us to formulate a (spatially/temporally) cohesive image, we rely heavily on the social conventions (causal chains) by which proper names are conveyed as the basis for our knowledge of the individual.

The new theory of reference argues that our knowledge of natural kinds has a similar basis. We encounter individuals of a natural kind, e.g., lemons, as glimpses or aspects of the entire kind. However, the knowledge we obtain by this direct experience would not, in general, suffice for us to know the absolute extent of the kind and/or distinguish its criterial properties.

How many people can distinguish a lemon from a yellow lime? Contrariwise, there are some green lemons growing in Brazil. Unless told otherwise, most people, even after detailed inspection, would probably mis-identify these as limes that simply have a somewhat different taste.

If I go to the store to buy lemons I rely heavily on their being labeled as such. On the other hand, even when the lemons aren't labeled, I usually get it right since it's the only small yellow object in the fruit section (since yellow limes and green lemons are rarely sold). Here I am using one of the characteristic (but not criterial) properties of lemons to select it from a limited range of alternatives. Further, the limitation to a few alternatives has been socially determined by the institution of supermarket, and the fruit section. I might not be so successful in the open jungle.

Putnam suggests that semantics, ultimately, depends on sociolinguistic considerations. In particular, the references of natural kind terms are seldom completely understood by people individually, but rather as a cooperative effort. He proposes a

> *Hypothesis of the Universality of the Division of Linguistic Labor:* Every linguistic community exemplifies the sort of division of linguistic labor just described; that is, it possesses at least some terms whose associated 'criteria' are known only to a subset of the speakers who acquire the terms, and whose use by the other speakers depends upon a structured cooperation between them and the speakers in the relevant subsets.
>
> ... We may summarize this discussion by pointing out that there are two sorts of tools in the world: ... there are tools like a hammer or screwdriver which can be used by one person; and there are tools like a steamship which require the cooperative activity of a number of persons to use. Words have been thought of too much on the model of the first sort of tool. [Putnam, 1970/1977:126-127]

F. NATURAL VS SOCIAL KINDS IN ORGANIZATIONAL VOCABULARY

In the philosophical literature, the term 'natural kinds' is used to indicate the (referents of) a wide range of natural language substantives, e.g., water, lemon, chair, house. Quine (1969) points out that for *some* natural kinds, e.g., water, lemons, there exists a scientifically accepted procedure of identification. For instance, chemistry defines water as the molecular compound H_2O, botany (I think) has a criterial definition for lemons or at least for lemon trees.

It is generally recognized that scientific explanation is ultimately a matter of social convention that changes as new theories are proposed (Kuhn 1962). We no longer accept the 'ether' as the basic substance of the universe. Likewise, it's conceivable (though not likely) that the scientific conception of water might change with further discoveries in particle physics.

But scientific explanation is a unique type of social convention in that it is ***authoritative***. What science accepts, the world accepts. H_2O is accepted as ***the*** definition of water because chemistry says so. Our informal conception of water includes water plus other impurities, though if disagreements arise, we generally accept the chemistry explanation as the criterial definition of 'pure' water.

This is not to say that all people who use the term 'water' understand its chemistry. Obviously, only a few do. Rather the semantics of this term rely on a social cooperation that leads ultimately to certain scientifically qualified individuals. A similar semantic cooperation exists in the common understanding of 'lemon,' which leads backwards from consumer, to supermarket, to farmer, to botanist.

The above remarks need qualification. Not all scientific paradigms are international in scope. Physics is, economics isn't. Further, even within a given society, the various scientific disciplines have differing epistemological status. For example, physics and chemistry seem to have more social credibility than psychology and sociology. This has important linguistic consequences for without this credibility our informal usage of terminology can have a different denotation than the scientific usage. For example, we accept as (pure) water exactly that which a chemist analyzes as H_2O. However, we do not for instance accept the meaning of 'anxiety' as what psychometrics measures using Galvanic skin response.

Very roughly, there seems to be an hierarchy of epistemological confidence within the physical sciences (e.g., physics, chemistry, astronomy) in uppermost status followed by the biological sciences (biology, botany, medicine), followed perhaps by psychology and then the social sciences (sociology, economics, political science).

The difficulty (Thom 1975, Berlinski 1976) is in the structural stability that can be assumed of the phenomena under study. We are comfortable with the assumption that physical phenomena are time/space invariant. Water is water whether on earth or on moon, in the eons past or those to come. Biological sciences have to consider evolutionary factors: fruit flies vary from one continent to another, bacterial diseases can evolve in a matter of months. The social sciences have an ever weaker claim to structural stability. Culture and social organizations obviously have enormous geographic and temporal variation. Psychology, acknowledging the effect of social context, suffers similar epistemological uncertainties. Indeed, even the presumed constancy of cerebral specialization may be culturally dependent (Sibatani 1980).

The importance of these observations here is in their linguistic consequences. The semantics of a given term is clearly a matter of social convention. However, that doesn't take us very far unless we can get some insight into the relevant socio-linguistic mechanisms. One of these mechanisms is the authority granted to scientific theories as defining the referents to certain of our terms.

As was suggested, the semantic problems of database translation and the verification of inference are likely to be least difficult for predicate terms that have a basis in the natural (physical, biological) sciences. More semantic instability is to be expected for terms signifying social artifacts. Consider again the term 'chair'. It seems doubtful that there can *ever* be a scientific explanation for this concept. Indeed, there are furniture design companies whose marketing strategy is to change our current conception of this term.

With respect to organizational databases, the vocabulary we are concerned about might relate to any of the scientific areas just mentioned. For example, databases relating to engineering, production or inventories may include terms based in chemistry (e.g., petroleum derivatives), physics (e.g., electronics engineering), and botany (e.g., agricultural inventories).

With respect to the vocabulary originating in these scientific disciplines, there is fairly wide semantic consensus and stability. The rate of linguistic change is likely to be slow relative to the time frame of the organization. Then consider the terminology relating to technological applications. In these areas the rate of linguistic change is much more rapid. It is, nonetheless, a fairly organized evolution. For example, trade journals and industry wide meetings and exhibitions help to standardize usage. To enable compatibility between products and processes, industry standards are eventually developed. This too helps to standardize usage.

However, a great deal of the vocabulary in organizational databases relates to socially defined phenomena. The relevant factor here is the social scope of the organization's interactions. An example is the relationship 'marriage.' In most cases, this can be accepted as a stable concept, relative to the interests of the organization. In other cases, e.g., the census bureau or cross-cultural organizations, linguistic variations have to be considered.

A more fundamental point is that the organization itself defines a social context and creates its own social artifacts. Prominent among these is its product offering, which, to be successful, is intentionally differentiated from related products in the marketplace. This product offering is furthermore dynamic, the effect of product development and marketing efforts. While the attendant linguistic change is managed within the organization, serious difficulties arise for e.g., regulatory, taxation, and consumer protection agencies.

The structure of the organization is itself a social artifact. This includes the identification of organizational substructures (divisions, departments, committees), organizational roles (manager, clerk), procedures, rules, standard documents, etc. These are described in a rich, locally defined, organizational vocabulary.

As the organizational structure and processes evolve in response to changes in the environment, this vocabulary must correspondingly evolve. If this vocabulary is used only for informal communications within the organization, this evolution continues naturally. However, as more and more of this vocabulary becomes embedded in the organization's structured information system, this linguistic evolution, hence the organization's adaptability, becomes restrained.

The difficulty follows from the remarks on the semantics of formal languages made in the last chapter. The design of a formal language depends on Frege's 'Principle of Compositionality'. That is, the semantics of a compound expression are constructed from the semantics of its syntactic constituents. Database queries and higher level inferences depend on this consistency for their validity. If the assumption of semantic stability is removed, the deductions provided by the information system can no longer be trusted. This issue is re-considered in the final chapter.

CHAPTER 5: THE SEMANTICS OF MEASUREMENT

CONTENTS

A. NUMBERS IN DATABASES

A large part of administrative data is numeric. Numbers differ from other types of data (e.g., character strings) in that we can perform arithmetic on them. Who says? On what basis do certain numbers convert into others (using e.g., addition, subtraction) with the supposition that these new numbers are meaningful? This is essentially the same type of question we have been asking with regard to the names of individuals and properties: what are the semantics of our use and manipulation of numbers?

The interface between mathematics and reality is *measurement*. Whereas a central issue in our discussion of individuals and natural kinds was *equality* (identity), the basic relationship reflected in numeric measurement is *order* or magnitude. An ordering relationship is any that can be described using the terms 'greater than,' 'less than,' and as well, 'equal to.' Examples are 'longer than,' 'heavier than,' 'more intelligent than,' 'prettier than,' etc. The objective of measurement is to translate these real world (empirical, observed) relationships into mathematical relationships that can be manipulated arithmetically.

The problem can be stated mathematically as establishing a correspondence between two systems of relationships, or *relational systems*, one applying to real world objects, the other to numbers. This is the subject of the emergent field of *measurement theory**, which is now briefly described.

---

* Not to be confused with *measure theory*, a topic in abstract algebra. Background references in measurement theory are Ellis (1966), Krantz, et. al. (1971), Adams (1979), Roberts (1979), Ghiselli, et. al., (1981).

B. MEASUREMENT THEORY

Measurement theory is based on the theory of relations, which in turn is based on set theory. Suppose $A_1, \ldots, A_n$ are sets. The *Cartesian product*

$$A_1 \times A_2 \times \cdots \times A_n$$

is the set of all ordered n-tuples $(a_1, a_2, \ldots, a_n)$ such that $a_1 \in A_1$, $a_2 \in A_2, \ldots, a_n \in A_n$. The notation A^n denotes the Cartesian product of A with itself n times. An ***n- ary relation*** on the set A is a subset of A^n. The number n is called the ***arity*** of the relation. A ***binary relation*** is thus a subset of $A \times A$, i.e., with an arity of 2.

Recall from the discussion on model theory in chapter [3] that a one place predicate denoted a subset of D, whereas a two place predicate denoted a binary relation on D, an n-place predicate denoted an n-ary relation on D. While, in the language, these are all represented as predicates, the difference between one and many place predicates is substantial from an inferential standpoint. The deductions one makes from the manipulation of simple subsets tend to be fairly obvious. However, the interplay between relations becomes increasingly more complicated as the arity increases. Even the characteristics of binary relations can be quite varied, as shown in Table [5.1].

Orders are special types of binary relations. Various types of orders can be defined, depending on their relational characteristics. See Table [5.2].

A Binary Relation (A, R) Is:	Provided That:
Reflexive	aRa, for all $a \in A$
Nonreflexive	it is not reflexive
Irreflexive	$\sim aRa$, for all $a \in A$
Symmetric	$aRb \rightarrow bRa$, for all $a, b \in A$
Nonsymmetric	it is not symmetric
Asymmetric	$aRb \rightarrow \sim bRa$, for all $a, b \in A$
Antisymmetric	aRb & $bRa \rightarrow a = b$, for all $a, b \in A$
Transitive	aRb & $bRc \rightarrow aRc$, for all $a, b, c \in A$
Nontransitive	it is not transitive
Negatively transitive	$\sim aRb$ & $bRc \rightarrow \sim aRc$, for all $a, b, c \in A$;
Strongly complete	for all $a, b \in A$, aRb or bRa
Complete	for all $a \neq b \in A$, aRb or bRa
Equivalence relation	it is reflexive, symmetric and transitive

Table [*5.1*]. Properties of Relations (from Roberts, 1979:15)

	Relation Type						
Property	Quasi Order	Weak Order	Simple Order	Strict Simple Order	Strict Weak Order	Partial Order	Strict Partial Order
Reflexive	X					X	
Symmetric							
Transitive	X	X	X	X		X	X
Asymmetric				X	X		X
Antisymmetric			X			X	
Negatively transitive					X		
Strongly complete		X	X				
Complete				X			

Table [*5.2*]. Order Relations (from Roberts, 1979:15)

Relations may be defined *extensionally*, by listing each of its n-tuples, or *intensionally*, by indicating an n-place predicate that selects the n-tuples. The latter method obviously requires the specification of the domain, called its underlying set, to determine its extension. For example, (Roberts 1979:16) let A be the set of all people in the world, and B be the set of all males in the world. Then,

$$R = \{(x,y) \in A \times A: \text{BROTHER}(x,y)\}$$

is different from

$$R' = \{(x,y) \in B \times B: \text{BROTHER}(x,y)\}$$

For instance R' is symmetric whereas R is not.

In some discussions, the term 'relation' is used to indicate the intensional predicate used to define it. The term 'relational system' is then used to indicate a relation in the extensional sense, and it is designated as a pair (A,R), where A is the underlying set and R is the defining predicate.

A ***function*** is of course a special type of relation, typically expressed as an (n+1)ary relation, denoted $f{:}A^n \rightarrow A$ such that

$$(\forall x_1, \ldots, x_n \in A)(\exists y \in A)[(x_1, \ldots, x_n, y) \in R],$$

$$(\forall x_1, \ldots, x_n, y, z \in A)[[(x_1, \ldots, x_n, y \in R] \;\&\; [(x_1, \ldots, x_n, z) \in R] \rightarrow y = z.$$

Functions of the form $f{:}A \times A \rightarrow A$ are called ***binary operations***, or just ***operations***. A ***relational system*** can be more generally defined as a set with various binary or higher arity relations (predicates) and operations (functions) defined over it. If there are p such relations and q operations, the relational system is a p+q+1 tuple:

$$RS = (A, R_1,\ldots,R_p, o_1,\ldots,o_q).$$

The *type* of a relational system is a sequence $(r_1, r_2, \ldots, r_p, q)$ where r_i is the arity of R_i (i.e., $r_i = m$ if R_i is m-ary).

For example, consider the measurement of mass. Let A be a set of discrete objects whose mass we want to measure. Let H be a relation on A, where H(x,y) indicates that x is heavier than y. Let $\oplus$ be an operation on A that yields the logical sum of two objects in A, e.g., $x \oplus y = z$. This is summation in the sense of Goodman's calculus of individuals where z is a third individual that has x and y as parts (though possibly geographically separated). Then $(A,H,\oplus)$ form a relational system of types (2,1). Note that this is a relational system defined over physical objects.

Consider the relational system RS = (Re,>,+). This is a numerical relational system, where Re is the set of real numbers, > is the numerical ordering relationship, and + is arithmetic addition.

As indicated earlier, the goal in measurement theory is to examine the correspondence between relational systems defined on real world phenomena such as $(A, H, \oplus)$ and numerical relational systems like (Re,>,+) for the real numbers. A mapping f from one relational system (RS_0) to another (RS_1) which preserves all the relations and operations of the first system is called a ***homomorphism***. More precisely, let

$$RS_A = (A, R_1, R_2, \ldots, R_p, o_1, o_2, \ldots, o_q)$$

$$RS_B = (B, R_1', R_2', \ldots, R_p', o_1', o_2', \ldots, o_q').$$

and we assume RS_A and RS_B are of the same type. A function $f: A \rightarrow B$ is a homomorphism from RS_A into RS_B if for all $a_1, a_2, \ldots, a_{r_i} \in A$

$$R_i(a_1, a_2, \ldots, a_{r_i}) \leftrightarrow R_i'[f(a_1), f(a_2), \ldots, f(a_{r_i})], \quad i=1,2,\ldots,p$$

and for all $a, b \in A$

$$f(aO_i b) = f(a)O'_i f(b), \quad i=1,2,\ldots,q.$$

In general, a ***fundamental measurement*** is recognized as a homomorphism from a real world (observed, empirical) relational system to some specified numerical relational system. This is distinguished from ***derived measurements***, which are computations based on fundamental measurements. (Table [5.4], for example, lists 63 derived measurements common in the physical sciences, and the six fundamental measurements on which they are based.) A fundamental measurement is fully described as the triple (RS_0, RS_1, f), called a ***scale***. This functional characterization of measurement is a relatively recent innovation over more conventional views. Adams ([1979, p. 210]) remarks:

> It is important to stress to students unfamiliar with the modern ***functional*** representation of measurement that fundamental measurement theory adopts this representation, and in so doing 'paraphrases' or 'translates' more traditional metrical language which is still widely prevalent in the sciences) into unfamiliar and initially unintuitive forms. In particular, metrical data are traditionally reported in ***denominate*** number form, e.g. as 'x weighs 5 lbs' (where '5 lbs' is a denominate number distinct from the 'pure number' 5), whereas the functional representation would paraphrase these reports into pure number form, e.g. as 'lb(x) = 5', where lb(x) is the ***weight-in-pounds*** function whose values are pure real numbers. In banishing denominate number ontological categories, the functional representation accomplishes an ***ontological reduction*** which is conceptually important.

Two basic issues that measurement theory investigates are the representation and uniqueness problems.

The ***representation problem*** is the following: for a particular relational system, RS_1, what are the necessary and sufficient requirements of the real world system RS_0, to guarantee the existence of a homomorphism, f? These requirements are called the ***axioms*** of the representation. Roberts comments:

> The axioms for a representation give conditions under which measurement can be performed. The axioms can be thought of as giving a foundation on which the process of measurement is based. In a less global sense, the axioms can also be thought of as conditions that must be satisfied in order for us to organize data in a certain way. In any case, it is important to be able to state such foundational axioms, at least for measurement in the social sciences. For we must know under what circumstances certain kinds of scales of measurement can be produced. In the physical sciences, the situation is different. We by now have well-developed scales of measurement, and writing down a representation theorem for these scales is often more a theoretical exercise than a significant practical development. (Roberts 1979:55)

The other basic issue of measurement theory pertains to the ***uniqueness*** of the homomorphism f. Suppose f is one homomorphism from a relational system RS_A into a relational system RS_B, and suppose A is the set underlying RS_A, and B underlies RS_B. Hence $f{:}A \rightarrow B$. Also suppose Φ to be another function mapping the ***range*** of f, i.e., the set

$$f(A) ::= \{f(a): a \in A\},$$

into the set B. Hence $\Phi{:}B \rightarrow B$. Thus the composition $\Phi(f(A))$ is a function mapping A into B into B.. If $\Phi(f(A))$ is also a homomorphism from RS_0 into RS_1, then Φ is an ***admissible transformation of scale.*** It is the admissible transformations that distingish the various scale types, e.g. nominal, ordinal, interval, ratio, absolute, as illustrated in Table [5.3]. Roberts (1979:55) remarks:

> ... a uniqueness theorem tells us what kind of scale f is, and gives rise to a theory of meaningfulness of statements involving scales. In particular, a uniqueness theorem puts limitations on the mathematical manipulations that can be performed on the numbers arising as scale values. ... [One] can always perform mathematical operations on numbers (add them, average them, take logarithms etc.). However, the key question is whether, after having performed such operations, one can still deduce true (or better, meaningful) statements about the objects being measured.

Admissible Transformations	Scale Type	Example
$\Phi(x) = x$ (identity)	Absolute	Counting
$\Phi(x) = \alpha x,\ \alpha > 0$ Similarity transformation	Ratio	Mass Temperature on the Kelvin scale Time (intervals) Loudness (sones) Brightness (brils)
$\Phi(x) = \alpha x + \beta,\ \alpha > 0$ Positive linear transformation	Interval	Temperature (Fahrenheit, centigrade, etc.) Time (calendar) Intelligence tests, "standard scores"?
$x \geq y$ iff $\Phi(x) \geq \Phi(y)$ (Strictly) monotone increasing transformation	Ordinal	Preference? Hardness Air quality Grades of leader lumber, wool, etc. Intelligence tests, raw scores
Any one-to-one Φ	Nominal	Number uniforms Label alternative plans Curricular codes

Table [*5.3*]. Some Common Scale Types (from Roberts, 1979:64)

C. MEASUREMENT SEMANTICS

The topic of measurement was introduced as the interface between mathematics and the world. Yet the theory of measurement, as suggested by the preceding sketch of its basic definitions and principles, is purely mathematical: it is the correspondence (functional mapping) between two relational, hence mathematical, systems. While the discussion of measurement theory generally presumes that one of these relational systems has an underlying set of real objects while the other underlying set is numbers, this is not required in its mathematical structure. Both could be real objects or both could be numbers. Measurement theory only recognizes them as sets. Elaborating this point, Adams (1979:211) remarks:

> several empirical systems associated with different types of fundamental measurement are often supposed to be representable in the same numerical system, and in such circumstances fundamental measurement theory commonly abstracts and considers the axioms which *any* empirical system must satisfy in order to be representable in the given numerical system, independently of the particular characteristics of the systems being represented. ... This abstraction from the special characteristics of, say, length addition as against weight addition (***which is unfortunate in some ways in drawing attention away from potentially important matters of empirical detail***) leads to what can by now be called the 'traditional' fundamental measurement categories: extensive, ordinal, interval, and so on, each of which is characterized by a particular numerical system in which varieties of empirical systems may be represented. Accepting this sort of abstraction, contributions to fundamental measurement theory commonly take the form of 'axiomatizations of extensive (ordinal, interval, etc.) measurement', rather than of weight, or temperature, or time measurement, etc. [emphasis added].

Here some of the same disappointments as with model theory are encountered. Assuming these underlying sets to be well defined *a priori* seems to wave away the key issues: how do we go about defining these sets, relations and operations? Whereas measurement theory concerns itself with mathematical requirements, here we are seeking something more, i.e., an epistemology of measurement. It is this epistemological aspect that gives us more confidence in for example physical measurement as opposed to psychological measurement (hence more confidence in physics than psychology as sciences).

Following the line of inquiry suggested by the 'new theory of reference' as an epistemological extension to model theory, an analogous approach might be tried here. Central to the new theory of reference was the notion of 'causal chains,' explaining the social mechanisms for the propagation of names of individuals and natural kinds through a linguistic population, both geographically and temporally dispersed. Consider how the various measurement concepts are propagated. Krantz et al. (1971:Ch.10) identify six fundamental measures from which nearly all the physical attributes that have ratio scale measures can be derived as simple monomials (see Table [5.4]). These are length, mass, time, temperature, electric charge and plane angle. As they point out, each of these is defined in terms of some natural phenomena or some standard object.

Length, for instance, is measured in terms of a certain meter length rod in Paris. Reproductions of this rod, subject to certain quality control requirements, propagate the measure 'meter' throughout the culture. Similarly, the gradations on the rod convey the concept of millimeter, centimeter, etc. These gradations convey a concept of *concatenation* of these smaller measures to the larger intervals. Length is measured by comparing an object to such reproductions of the standard rod. When the extent of the object is more than one rod, we can concatenate several such rods to form larger measures. Indeed, we may consider length to be *defined* by these operations.

Concepts of mass and time, also involve some standard object —i.e., standard kilograms (etc.) and the standard interval of a certain pendulum in Washington, D.C. Here too there is an analogous concept of concatenation: for instance, concatenation of two pendulum swings doubles our measurement.

Measured phenomena that include a concept of concatenation are called *extensive attributes*. They have the following properties (where x and y are objects, a and b are numbers):

a) of ordering $x \, R \, y \leftrightarrow f(a) > f(b)$

b) of additivity $f(x \, O \, y) = f(a) + f(b)$

The ordering relation, R, is assumed in all types of measurement. In the second additivity assumption, the concatenation operation, O, has its counterpart in arithmetic addition.

Phenomena where the concatenation operation may not be interpreted as arithmetic addition are called *intensive attributes*. Examples are density and temperature. Here the arithmetic concept of additivity is constructed by means of a more complex experimental operation.

This illustrates that the scale type employed is to a certain extent a matter of the state of scientific experimentation and instrumentation. Roberts notes from Stevens (1959:124) in the case of temperature: early man probably only distinguished hot and cold. Later, comparative terms were probably introduced, e.g., hotter, colder. The invention of thermometers led to internal scales of temperature. Finally, the development of thermodynamics led to a ratio scale of temperature, the Kelvin scale.

But, what is it that leads us to more epistemic confidence in, say, length measures than intelligence measures. If we can accept length to be defined operationally as what we measure with a meter stick, why are we less satisfied with the psychologists' explanation that 'intelligence is what an IQ test measures'. The argument sometimes given is that physical measures are 'time-space invariant' whereas e.g., psychological measures are not. But what is it that is supposed to be time-space invariant? If I measure Ronald Reagan's height in summer 1982 in Rome and you measure Ronald Reagan's height in Helsinki during winter 1984, the two measures would roughly agree, modulo random measurement error plus perhaps a minor downward bias due to the cold and the slouch of increased age. However, if instead of measuring height we gave RR an IQ test at each of these two places and times couldn't we also expect a rough accordance of results, modulo random error plus perhaps a minor downward bias due to aging? Why is it that we object to certain IQ tests having racial, cultural, etc. biases and hence not 'really' measuring intelligence?

Consider our measures of time. We have various *objective* measures based on our assumed regularity in the behavior of such objects as sundials, hour glasses, mechanical and digital clocks. Yet we also have a *subjective* sense of time that does not always accord with the objective one — e.g., 15 minutes in a dentist chair may seem much longer than an hour at an exciting party. We seem to put more confidence in the objective measure of time —why?

What we call 'confidence' here appears to be more social pragmatism. We need to coordinate temporally with other people, e.g., trains, buses, school and working hours, TV programs, doctor's appointments. We (apparently) have different subjective measures of time, so we need some non-human process that we can all agree on. Originally it was the passage of the sun in the sky, then a certain pendulum, now it's molecular vibrations. However, in these cases it is not the fact that time is mechanically computed that makes it objective, but rather that we have a criterion of sameness (similarity) in the computation. When two bank robbers synchronize their watches, they don't so much care that they indicate the correct time as the same time. One could imagine a primitive, isolated, cloudy-skied village where one individual was selected as the official time maker: the pronouncements of his subjective sense of time passing would serve as the local standard.

Note the similarities here with the social view of semantics developed in the last chapter. Whereas in medieval times everyone could individually watch the sun and recognize for example high noon, our current perceptions of time require a much richer complex of technology and scientific expertise. Like the way I understand the concept of 'lemon,' my understanding of time depends as well on social cooperation. That is, the basis of our objective agreement is no longer something we all know directly (i.e., the sun), but something we rely on other people (scientists, production engineers in the manufacture of watches) to know.

Similarly with other physical measures such as length, mass and temperature, our concept of what these are has come to rely heavily if not entirely on scientific measurement as the definitional authority. For example, if we feel a chill but the weather broadcast announces 75°F (or 20°C) we tend to doubt our health rather than the weather reporting system. That is, our internal or subjective sense of temperature doesn't count for much epistemologically in comparison to the external, objective measure.

But the authority of a particular measurement standard relies also on its means of communication throughout the society. This depends on the technology for reproducing, operationally, similar imitations of the original standard. If for example everyone's watch went arbitrarily faster or slower than the standard, these mechanical measurements would be no better than our subjective senses of time.

But what about, for instance, intelligence measures? Why do we have less epistemic confidence in these cases? Isn't it that our own subjective sense of intelligence counts more heavily? For instance, we may have our own internal ranking of the intelligence of those we know. If we hear the IQ scores of those people, and they disagree strongly with our subjective measures, we in this case doubt the IQ test as a measure of intelligence.

We can't doubt the weather bureau because the scientific measure of temperature *is* our concept of temperature. We accept science's authority in defining the concept of temperature, and we have confidence in the technological reproduction of temperature measurement devices. The problem in social science measurement (psychometrics, sociometrics, econometrics) is that there are intuitive, subjective concepts of the attributes being measured, which we take as epistemologically stronger than objective measures. Consider, for instance, psycho physics: objective measures such as perspiration are used as proxy measures for subjective states, e.g., anxiety.

The problem in social science measurement is that we have two competing standards, what we have called subjective and objective. Since our subjective sense seems to us the more sound, we expect objective measures to conform to it. In the case of physical measurement, we no longer attach this importance to subjective measures. Consider for example the lack of scientific interest in building a clock or a thermometer that measured our subjective senses of time or temperature.

Quality	Exponents of Q	Θ	M	L	T	A
Base quantities						
Charge (electric)	1					
Temperature		1				
Mass			1			
Length				1		
Time					1	
Plane angle						1
Kinematic (L, T, A) quantities						
Curvature				-1		
Wave number				-1		1
Angular acceleration					-2	1
Time constant					-1	
Angular velocity					-1	1
Frequency					-1	1
Plane angle						1
Solid angle						2
Period					-1	-1
Time					1	
Acceleration				1	-2	
Acceleration of gravity				1	-2	
Velocity, speed				1	-1	
Velocity of light				1	-1	
Wave length				1		-1
Length				1		
Diffusion coefficient				2	-1	
Kinematic viscosity				2	-1	
Area				2		
Volume velocity				3	-1	
Volume				3		
Mechanical (M, L, T, A) quantities						
Rotational compliance		-1	-2	2	1	
Rectilinear			-1		2	
Specific refraction			-1	3		
Acoustic capacitance			-1	4	2	
Acoustic impedance			1	-4	-1	
Acoustic resistance			1	-4	-1	
Acoustic reactance			1	-4	-1	
Inertance			1	-4		
Density			1	-3		

Table [*5.4*]. Dimensions and Units of Physical Quantities.
(from Krantz, et al. 1971:539-544)

Quantity	Exponents of					
	Q	Θ	M	L	T	A
Energy density			1	-1	-2	
Pressure			1	-1	-2	
Stress			1	-1	-2	
Modules of elasticity (Young's)			1	-1	-2	
Bulk modules			1	-1	-2	
Tensile strength			1	-1	-2	
Shear modules			1	-1	-2	
Shear strength			1	-1	-2	
Viscosity			1	-1	-1	
Sound intensity			1		-3	
Poynting vector			1		-3	
Surface tension			1		-2	
Mechanical rectilinear resistance			1		-1	
Mass			1			
Force			1	1	-2	
Momentum			1	1	-1	
Impulse			1	1	-1	
Radiation intensity			1	2	-3	2
Power			1	2	-3	
Energy, work			1	2	-2	
Quantity of heat			1	2	-2	
Moment, torque			1	2	-2	
Mechanical rotational resistance			1	2	-1	-1
Action			1	2	-1	
Angular momentum; moment of momentum			1	2	-1	
Moment of inertia			1	2		
Thermal (Θ, M, L, T) quantities						
Heat capacity (mass)		-1		2	-2	
Heat capacity (volume)		-1	1	-1	-2	
Thermal conductivity		-1	1	1	-3	
Entropy		-1	1	2	-2	
Molar gas constant		-1	1	2	-2	
Temperature gradient		1		-1		
Temperature		1				

Table [5.4] continued.

Quantity	Exponents of Q	Θ	M	L	T	A
Electrical and magnetic (Q, M, L, T, A) quantities						
Permeability	-2		1	1		
Impedence (electric)	-2		1	2	-1	
Resistance (electric)	-2		1	2	-1	
Reactance (electric)	-2		1	2	-1	
Coefficient of inductance	-2		1	2		
Permeance	-2		1	2		
Resistivity	-2		1	3	-1	
Magnetic induction; magnetic flux density	-1		1		-1	
Electric field intensity	-1		1	1	2	
Vector potential	-1		1	1	-1	
Potential (electric)	-1		1	2	-2	
Electromotive force	-1		1	2	-2	
Magnetic flux	-1		1	2	-1	
Quantity of magnetism	-1		1	2	-1	
Flux linkage	-1		1	2	-1	1
Magnetic moment	-1		1	3	-1	
Charge density (volume)	1			-3		
Current density	1			-2	-1	
Pole density	1			-2	1	
Electric displacement	1			-2		
Polarization	1			-2		
Magnetic field intensity	1			-1	-1	
Magnetization	1			-1	-1	
Sheet current density	1			-1	-1	
Linear charge density	1			-1		
Current	1				-1	
Magneto-motive force	1				-1	
Charge	1					
Flux (electric)	1					
Pole strength	1			1	-1	
Dipole moment	1			1		
Magnetic (dipole) moment	1			2	-1	
Conductivity	2		-1	-3	1	
Permittivity	2		-1	-3	2	
Reluctance	2		-1	-2		
Admittance (electric)	2		-1	-2	1	
Conductance	2		-1	-2	1	
Susceptance	2		-1	-2	1	
Capacitance	2		-1	-2	2	

Table [5.4] continued.

CHAPTER 6

THE SEMANTICS OF ACCOUNTING

CONTENTS

A. THE OBJECTIVES OF ACCOUNTING

Accounting has long been a major source of management information. Indeed, it is often called the 'language of business'. The history of financial record keeping probably dates from the first uses of currency. Our 'modern' double entry system of accounting itself has a respectably long tradition. The first text on double-entry bookkeeping was published by Pacioli in 1494, though Mattessich points out that the Genoese were using a similar system as early as 1340.

Ijiri (1975:177) points out that "the basic double-entry scheme of bookkeeping has remained unchanged over the past 500 years. Considering the enormous changes in economic systems during these years, the stability of the double entry system is truly astonishing."

Recognizing the phenomenological point that we tend to perceive the world through our representations of it, he points out that this framework is largely responsible for our present conception of profit and indeed perhaps of capitalism itself. He quotes (1967:109) Sombart (1928):

> One can scarcely conceive of capitalism without double entry book-keeping: they are related as are form and content. It is difficult to decide, however, whether in double-entry bookkeeping capitalism provided itself with a tool to make it more effective, or whether capitalism derives from the 'spirit' of double-entry bookkeeping.

Hence, simply by the weight of its historical acceptance, accounting is clearly more than an arbitrary choice for representing business activity. Yet, in recent years, this same tradition has been criticized by economics and other decision sciences as being overly dogmatic and ritualized, "based on empty identities, as being concerned with trivial problems, as propagating unscientific methods, as hampering progress in business administration and economics" (Mattessich 1964:104).

Why should this long success suddenly come into doubt? There are several interrelated factors. Accounting's primary function traditionally has been a custodial one, based on a separation of capital ownership from the management of operations. This has two aspects

a. documentation: maintaining detail records of each legal/economic transaction

b. evaluation: summarizing these transactions into an overall concept of wealth and profitability.

Thus accounting's primary function, historically, has been reporting to external audiences having a financial claim on the firm: investors, lenders and tax agencies. For this reason, accounting standards are established by independent agencies (e.g. the Financial Accounting Standards Board in the US) and the organization's accounting reports prepared for these external parties are verified by externally certified public accountants.

Yet, apart from these external audiences, accounting is also the primary information source for internal management. It is here, for purposes of managerial decision making, that criticism has been sharpest. One line of criticism is that accounting does not adequately distinguish the needs of internal management from external investors. Hence internal management reports tend to be more detailed variants of the external reporting model.

The question, then, is why should management accept inadequate reporting given that they can direct resources to produce reports as they want them? The answer, probably, is that since external reports are the basis for evaluating the firm's performance, these measures become the proxy goals that management seeks to optimize.

At present, at least in western countries, the principal external reports are the balance sheet, funds flow (sources and uses of funds), and income statement. For private firms, the income statement, particularly the profit calculation, receives the most attention, for this is regarded as the best index of the firm's progress. Thus, accounting has a central role in directing the efforts of private enterprise.

A key requirement, from the investor standpoint, is the *comparability* of financial accounting reports between firms. Hence, there is strong pressure to develop a standardized structure in these reports. However, the advantage of any particular firm is likely to be found in its *differentiation* from other firms. The knowledge or craft that a firm develops for its production and administration is called its *technology* (Woodward 1978). The internal reporting of the firm needs to reflect this technology in the types of entities that are recognized, basic organizing concepts and types of measures. The correspondence between the highly differentiated information of internal operations and the standardized structures required for external reporting present dialectical pulls on accounting theory and practice.

B. ACCOUNTING PRINCIPLES

The objectives of accounting were summarized as twofold: custodial and decision oriented. This raises the issue of the form that accounting theory should take to support these objectives.

For its custodial objectives, the emphasis is on verifiable evidence, protection from fraud, and the arbitration of potential conflict. Here the orientation is legalistic, stressing the detailed and organized recording of transactions with external parties (bookkeeping) and their verification and comparability (auditing). The concern is with the interpretation and objective summarization of past events.

This leads to the development of reporting standards, so-called 'Generrally Accepted Accounting Principles'. The difficulty is to develop rules that present the wide variation of different organizations' activities in a uniform reporting structure. While it is the 'custodi'al function that is most highly developed in accounting, there is nonetheless dissatisfaction with current theory in this area. Chambers(1973:46) notes:

> There was a time when people debated the character of 'accounting principles'. The debate was not particularly fruitful. It did not lead to a definite body of self-consistent rules. Grady's *Inventory of Generally Accepted Accounting Principles* [1965] and exercises of a similar kind in other countries have all left open a variety of optional methods of accounting for the same kinds of assets, liabilities, revenues and expenses. The directors and managers of companies are free to choose, from the options, such particular rules as will serve their purposes. Companies have switched from FIFO to LIFO and back again, from straight-line to accelerated depreciation and back again. If there were a firm body of rules, switching of these and other kinds would long since have been outlawed.

The problem here, perhaps, is that increasing rates of technological change and innovation lead to increased variation in organizational structures and activities. Accounting must continually revise its standards to accommodate innovative forms of organization.

Accounting principles are generally decided by national committees such as the Financial Accounting Standards Board (FASB) in the U.S. However, these typically do not have legislative power, but rather attempt to summarize and codify trends in accounting practice. Independent auditing firms accept these pronouncements as advisory, but exercise their own judgement in particular circumstances.

Focusing then on the auditing firm as the leading edge in the evolution of accounting standards, the question arises as to how practice amongst these companies is influenced, in particular the role of thoeretical research. Watts and Zimmerman (1976) remark:

> ...it is generaly concluded that financial accounting theory has had little substantive, direct impact on accounting practice or policy formulation despite half a century of research. [273]
>
> ... Understanding why accounting theories are as they are requires a theory of the political process. [275]

Somewhat cynically, they continue:

> Most theorists probably believe that an objective of their research and the reason they supply theories is to provide knowledge which will ultimately improve accounting practice. They would not regard themselves as supplying "excuses". But we suggest that the predominant contemporary demand for accounting theories (the demand for accounting in a regulated economy) is the demand for justifications —"excuses". [285]
>
> ... Thus, research and consulting funds will tend to flow to the most eloquent and consistent advocates of accounting practices where there are vested interests who benefit by the adoption or rejection of these accounting practices. [287]

However, the term 'theory' merits examination. Theories are not simply academic pronouncements. Theories are sets of propositions about (some aspect of) the world that are claimed to be true. Descriptive theories describe and predict actual phenomena. Normative theories prescribe behavior relative to some performance criterion. Accounting, as practiced, is not a theory in either of these senses. It is rather a doctrine, representing a particular value structure.

But what of the other, decision oriented role of accounting? Though separate, this is not independent of accounting's custodial and legalistic functions, as these provide the ultimate criteria of managerial performance. Unlike the custodial role, which is oriented to the documentation of past activities, the decision oriented role of accounting is necessarily future oriented. We don't decide the past, though we may have varying interpretations of it. Managerial decisions are rather decisions between alternative possible futures. However the link between these two roles of accounting is that the future that eventually occurs will be reported according to Generally Accepted Accounting Principles. Managers may have other goals and values not reflected by these reporting methods. However the prominence of accounting reports makes the goals and values they embody at least an important component in directing organizational performance.

On the other hand managers are concerned with more than the data in accounting reports. Their responsibilities are to direct the allocation of material resources and labor, to gauge the marketplace and the competition, and so on. In short, they deal with the realities behind the numeric valuations, i.e. the real world phenomena described in the language, L_{RW}. But here a theoretical gap is encountered: how are actual entities and activities in the organization related to accounting's evaluation of them? This is explored in the sections to follow.

C. ACCOUNTING MEASUREMENT AND VALUATION

As noted, the custodial role of accounting is concerned primarily with documenting and interpreting past activities, whereas for decision making the concern is to evaluate future possibilities.

The word 'evaluate' is key. In its decision support role, accounting is not concerned with the identification of future alternatives, but rather in assessing their financial impact.

The decision makers involved are typically taken to be internal managers or external investors. Other interested parties might also be trade unions, regulatory agencies, economic analysts, etc. These various parties obviously have differing interests with respect to the organization's activities. The manager is concerned with internal operations, departmental performance, product sales, etc. The investor is concerned with aggregate profitability of the firm with respect to others in his/her portfolio.

The point is that evaluation differs from objective description in that it comprises a value structure. In science, the dominant form of description is measurement, i.e. the mapping from a real world (objective, observable) relational system to a numeric one. It is intended that the measurement process be time/space invariant, i.e. that any two parties describing like phenomena should arrive at identical measurements.

Evaluation is however interest relative. Two parties may evaluate the same phenomena differently. This is closely related to the economic concept of utility. With respect to the pragmatic purposes of decision makers, however, utility theory is inferentially too weak. This is due to its measurement theoretic foundations. Utility theory offers only ordinal measurement*. That is to say, several items might be ordered in terms of their utility, e.g. A ⧁ B ⧁ C

indicates that C has more utility than B which has more utility than A. However, there is no corresponding concept of additivity, e.g. we have no basis for inferring that A and B together have more (or less) utility than C. Utility theory is therefore not an extensive measurement. The difficulty this creates is that it has no concept corresponding to arithmetic addition, hence does not permit numeric summarization.

Accounting, by contrast, might be viewed as a compromise between the ideals of utility theory and the inferential capabilities of extensive measurement.** Recall that in extensive measurement, the form of the numeric relational system had the form:

$$RS_1 = (Re, >, +)$$

i.e. arithmetic operations on the real numbers. The base relational system, what is being modeled in the arithmetic system, must therefore have the form,

$$RS_0 = (D, ⧁, \oplus)$$

where D is the set of phenomena being measured, ⧁ is an ordering on these phenomena, and $\oplus$ is a concatenation operation. Here ⧁ would be a preference ordering, whereas $\oplus$ would be an aggregation operation, analo-

* This is a complicated issue that in fact motivates much of the interest in measurement theory. See for example Krantz et. al. (1971), chapter 8 and Roberts (1979), chapters 5 through 8.

** See for example Ijiri (1967), Ijiri (1971), Chambers (1972), Ijiri (1975), Ashton (1977).

gous to that suggested in Goodman's calculus of individuals (discussed in chapter [3]). Thus, two objects taken together are always considered to be worth more than each one separately.

As noted, preference orderings are interest relative. One person's or organizations's preferences do not match another's; otherwise there would be no commercial exchange. For instance in a sales transaction, the money received by the seller is valued more than the good sold; likewise the purchaser values the good more than the money paid. Insofar as the currency is a stable and widely used medium of exchange, the price paid for an object is a conservative estimate of that object's worth to the purchaser.

Recall that in ordinary extensive measurement, the function mapping from RS_0 to RS_1 had only one argument, indicating the object being measured. In the case of accounting valuation, an additional argument is needed to indicate the entity for which the valuation is made. Thus the function, **V$**, which accounting adopts to map between these two relational systems, has the form:

$$\mathbf{V\$}\,(d, e) = n$$

where d is an element of D, i.e. the set of objects (services, etc.) being valuated, and e is the accounting entity. **V$** is assumed to be specific to a particular currency, e.g. U.S. dollars. Accounting theory amounts to an elaboration of the nature of the valuation function, **V$**. For instance, for monetary objects (cash, negotiable securities), the value n is given as the face value of the object. Inventory items are valued at cost (i.e., face value of the money paid for them) on either a FIFO or LIFO basis. Capital equipment is valued at historical cost reduced periodically through a time dependent depreciation algorithm.

Reflecting the parallels to extensive measurement, Ijiri (1967) notes that balance sheets need not be restricted to the typical asset and equity classification but for instance could have additional columns classifying assets by their geographical location or age (see Figure [6.1]).

Asset		Equity		Location		Age	
Cash	$10	Payables	$20	Head Office	$30	Under 6 mo.	$40
Receivables	20	Accruals	10	Factory	40	Under 1 yr	10
Inventories	20	Loans	40	Warehouse	30	Under 2 yr	10
Buildings	40	Capital	30			Over 2 yr	40
Equipment	10						
	$100		$100		$100		$100

Figure [6.1]. (source: Ijiri 1967:105)

Clearly, the assets could be grouped and measured under any number of such classifications schemes. Accounting transactions would thus be 'multi-entry' rather than just double entry.

It would seem a rather obvious extension to have alternative asset classifications according to managements interests and decision needs. Why then have accounting systems been so uniformly dedicated to the double-entry view?

The explanation Ijiri offers is that the double-entry method reflects a causal relationship:

> what makes the double-entry system double is not the double classification (Assets = Equity) that is often described in accounting literature but rather the cost principle, which recognizes the causal relationship between an asset acquired and an

> asset foregone. In this sense, the double-entry system and the historical cost principle...have a logical connection since one is a form developed to express the other. (Ijiri 1967:107-108).

Rather than explain the dual balance sheet classification, these remarks focus more on the relationship between revenues and costs, what in textbooks is sometimes called the 'matching concept'.

Accounting may therefore be viewed as having two dualities: assets vs claims on assets (equities); and the causal matching of revenues to the costs incurred in generating those revenues.

The two distinctions are orthogonal: assets vs equities reflects the difference between objects (collectively, wealth) and their ownership; the revenue vs cost distinction reflects efforts to increase wealth.

This double duality is reflected in the basic financial accounting reports, namely the balance sheet and income statement. The balance sheet portrays the financial position of the firm at a point in time whereas the income statement reflects changes in financial position over the interval of time between balance sheets. Actually this is only partially true, for the income statement reflects changes due to sales (i.e., the normal operations of the firm) and so is conceptually a sub-account of stock-holders equity (retained earnings). Other financial changes, e.g., acquisition of capital assets, new issues of bonds or stocks, are not included in the income statement and so motivate additional reports on the sources and uses of funds or working capital.

What this structure does, however, is impose one particular interpretation of the states and changes of the firm. This interpretation has come to be authoritative in specifying the claims of lenders and investors.

This links back to accounting's custodial role and its consequent adherence to historical cost measurement. Economics has long ago discarded historical costs as relevant for decision making. Critics of accounting, believing that accounting should adapt its practices to recognize decision making needs, have suggested price-level revaluations, replacement cost, market value, and other alternatives. Yet accounting theory reacts with austere conservatism. Indeed, conservatism is another of the textbook principles of accounting. An example of this would be the accounting principle to value inventories at the lower of their historical cost or their current market price. The point is that in *specifying* the claims of various stakeholders on the firm and its assets, conservatism reduces potential conflict. Good fences make good neighbors.

Thus, in its custodial role documenting the organization's activities, accounting not only describes but as well evaluates these activities. This evaluation serves a legal function in arbitrating the interests of claimants, say, in cases of liquidation. Accounting differs from simple measurement, therefore, in that it comprises a value structure. Through the weight of historical tradition, investors and managers alike have come to accept accounting as a (the?) principal criterion of corporate performance.

D. ACCOUNTING ONTOLOGY

Since the 1950s, there has been a somethat intermittent line of research to discover the basic logical foundations of accounting.*

More recently, the heavy impact of automated data processing on accounting systems has renewed interest in this foundational research to formalize accounting theory for computational inference. Importantly, as database management has developed, another line of research has developed relating accounting theory to the emerging data models**. Preceding chapters have repeatedly argued that databases, ignoring their mechanical aspects, are essentially formal languages. It is here that the theoretical connection between accounting and information technology is to be found.

In the last section it was argued that valuation was a fundamental aspect of accounting. Of central concern was the valuation function,

$$V\$(d, e) = n$$

which mapped a domain of objects, D, depending on the interests of the accounting entity, e, to a number, n. It is instructive to consider the domain of this function, that is the sorts of things subjected to accounting valuations. This leads us to an *ontology* of accounting, that is, the fundamental objects recognized in the theory.***

---

* For example, Chambers (1955), Mattessich (1957), Moonitz (1961), Mattessich (1964/1977), Chambers (1966/1975), Ijiri (1967/1977), Sorter (1969), Johnson (1975), Ijiri (1975), Caspari (1976), Hughes (1978), Carlson and Lamb (1981). Mattessich (1980) provides a historical review.

** For example, Lieberman and Whinston, 1975, Haseman and Whinston, 1976, Everest and Weber, 1977, McCarthy, 1979, McCarthy, 1982.

*** This is in the spirit of Quine's remark (1953/1961) "to be is to be the value of a bound variable." Ontlogy, it will be recalled from Chapter [1], asks the question "what is there?"; i.e., it is the nature of the entities forming the domain of individuals. Semantics, by contrast, is the mapping from expressions of the language into this domain.

The importance of ontology is to provide a compatible basis for various formalizations of accounting inference. Differing views of accounting can only be reconciled if they agree on what are the fundamental objects of the theories. What follows is to be regarded as an initial proposal. (This ontology formed the basis for the representation language, CANDID, presented in Lee, 1981.)

The concern here is to identify the types of individuals subject to accounting valuation. Closely related is the problem of individuating these entities, i.e., of consensual recognition and naming (discussed in chapter [4]).

Reviewing briefly, the view offered by Strawson (1959) was that the general basis for such identification is the locatability of these entities in a spatial/temporal framework.

The problem of individuation becomes especially important when we consider contractual objects like notes, bonds, stocks, options, licenses, insurance policies, etc. Clearly it is of critical importance for a company to know if it has a certain right or obligation. Indeed it is precisely because of this problem of identification that signed documents play such an important role in contractual transactions: the signed document represents the agreement in a form locatable in space and time. (This point is elaborated in the next chapter.)

Referring back to the accounting function, V$(d, e) = n, the entities in the ranges of d and e are designated, respectively, as *economic objects* and *economic objects*. If we consider only physical objects as economic objects and persons as economic actors, the ontological problem is trivial: both types of entities are locatable in space and time.

However, another common type of economic actor (at least in western societies) is a corporation. A corporation is more problematic from this perspective since it has no essential physical reality: no one of its assets, including its buildings, nor any one of its employees nor any of its executives or board members nor any one of its stockholders is essential to the identification of the corporation. Any one of these may change or be removed from the corporation, and the identity of the corporation can still continue.

The objects of economic activity, i.e., the things that are traded, present analogous ontological problems. Money for instance is a key object of exchange. Yet money is no longer uniquely represented by physical objects such as coins and bills, but often appears merely as magnetic records in bank accounts. These, like computer programs, lose the easy location in a unique place at a given time.

Information objects, such as recorded music, printed texts and computer programs were already mentioned as presenting a problem for identification. Such objects present an interesting legal problem in that they can be "stolen" (copied) without removal of the original. (Our notion of theft is basically a physical one.) Computer, communications and photocopy technology are bringing the characteristics of this type of object to prime economic importance.

One other type of non-physical economic object was also already cited: contractual objects. Signed documents have historically provided these types of objects with an easy physical identifiability. However, in most centers of trade in contractual objects, namely commodity, bond and stock market exchanges, there is a definite move towards automation of records and transactions, so that here too the identifiability of such objects becomes problematic.

Underlying the accounting valuation function V$ is a concept of ownership. To indicate that what we are exploring here might be a specialized version of ownership, somewhat different in accounting than in other language contexts, we denote it as a predicate:

OWN(d,e)

meaning that e, an economic actor, *owns* d, an economic object. The wealth of an organization, i.e., its assets, are the things that it owns. The accounting valuation function, V$, applies to just these things.

Another relationship between economic actors and objects is that of possession, written

POSS(d,e)

indicating that actor e possesses object d. Again, we recognize that the understanding of this concept may be specialized in accounting.

Intuitively speaking, ownership constitutes a set of rights granted by the legal system of an actor towards an object. Possession on the other hand refers to physical custody. Usually, an actor possesses what it owns, but not always, as in the case of loans and rentals.

E. ECONOMIC ACTORS

Persons, Proprietorships

The most obvious type of economic actor is an individual person, designated as:

$$PERSON(x).$$

However, in U.S. law, not all persons qualify as legitimate economic actors — minors and the insane are excluded. This more restricted set is designated LPERSON (legal person), defined as:

$$LPERSON(x) ::= PERSON(x) \ \& \ AGE(x,YR) \geq 18 \ \& \ SANE(x).$$

Personal businesses, owned by a single individual, are called *proprietorships*. In US law they are not distinguished from their owner, hence

$$PROPRIETORSHIP(x) ::= LPERSON(x)$$

Joint Ownership, Partnerships

Joint ownership is where one or more parties share equally in the ownership of an object. Essentially, the group of owners form a set that, as a unit, owns the object. For instance, for joint owners $x_1, \ldots, x_n$

$$OWN(z,y) \ \& \ z = \{x_1, \ldots, x_n\}$$

In US law, a partnership is an economic actor consisting of such a set of equally participating persons. Hence,

$$PARTNERSHIP(z) ::= \exists x_1, \ldots, x_n \ LPERSON(x_1) \ \& \ \ldots \ \& \ LPERSON(x_n) \ \& \ z = \{x_1, \ldots, x_n\}$$

Private Corporations

It is at this level that the concept of an economic actor becomes philosophically challenging. A corporation is an artifact of the legal system. In the US, it is a 'legal entity', entirely separate from and independent of its owners. Unlike proprietorships and partnerships, which are formed simply by the volition of the parties involved and have no separate legal status, a corporation is formed by a specially granted permission from the state.

Informally, this process is as follows. The group of people who want to start the corporation, called its *promoters*, submit registration information, called *incorporation papers*, and a *prospectus*, which describes the capital structure and intended function of the corporation to the governing state. If the corporation is to engage in interstate commerce, the prospectus must also be approved by the Securities and Exchange Commission (SEC).

In addition, a *certificate of incorporation* is filed by the promotors, which, if approved, is maintained by the office of the secretary of the state of incorporation. This certificate lists the corporation's principal offices, names of directors and incorporators, the total number of stock shares (each at a common value called the *par value)* and the name and number of shares held by each stockholder. The corporation cannot sell more than this initial number of shares without obtaining additional permission from the state. On acceptance by the state, this certificate becomes the corporation's *charter*.

This charter is a contractual permission by the state which, in gross terms, says the following: Stockholders have a right to vote members of the ***board of directors*** (at least three people) of the firm and to participate in the division of residual assets on the dissolution of the firm.

The board of director's main responsibility is to appoint *officers* of the corporation, who serve as the agents of the corporation in legal transactions (e.g., engaging the corporation in contracts, hiring and management of employees).

Only the officers, and the people they employ, can engage in the direct operation of the firm. Note that being a stockholder does not carry the right to participate in the management of the corporation nor to act as its agent in contracts.

To summarize, a corporation is essentially a locus of ownership, on one hand, and a locus of contractual commitment on the other. (These will define the two sides of the corporate balance sheet: its assets and its liabilities, including stockholder equity.) Changes in the things owned by the corporation and its commitments to other parties are made by the corporate officers and their employees, acting as agents. Corporate officers are appointed by the Board of Directors, who in turn are elected by the stockholders.

A crucial issue from a formal standpoint, however, is the identification of this locus of ownership and commitment. If we simply dismiss it as an 'abstract object' having no spatial/temporal location, we are left with the theoretical as well as very pragmatic problem of determining when the corporation exists and the boundaries of its rights and obligations.

However, as noted above, the critical event in the formation of a corporation is the granting, by the secretary of the state of jurisdiction, of the corporate charter. This provides the creation of the corporation with a unique location in space and time. Furthermore, the corporate charter provides the corporation with a unique *corporate name* (within that state). This provides any subsequent contracts and titles of ownership with a reference to the corporate charter, and hence to a unique spatial/temporal location.

Though this provides the means to identify a corporation, we have still not explained what a corporation *is*. Clearly, it is not in itself something physical. Rather it is a complex of contingent rights and privileges as established by the corporate laws of the state.

Let us refer to this complex as CORP-RIGHTS. These are granted by a particular state, and associated to a unique (within the state) corporate name.

We would like to say that the corporation is simply this permission. However, if we are speaking of a certain time, t, the corporation is not simply this permission at time t but, to account for the corporation's ownership of assets, it must also include permission at previous times when the assets were acquired. Further, while the corporation is in operation it will presumably have contractual obligations to other parties. These involve evaluation of these corporate rights not only in future times but under alternative circumstances, i.e., in other possible worlds.*

* In Lee (1981) the formal semantics of this is regarded using the Montague intension operator applied to the deontic permission.

The preceding discussion has dealt with *private corporations*, i.e., those that are profit oriented and have stockholders who ultimately receive these profits either through dividend distribution or dissolution of the corporation and sale of its assets.

Other types of corporations might also be described similarly. For instance, *non-profit corporations* do not have stockholders nor do they pay income tax. *Quasi public* corporations are private corporations that provide certain public services (e.g., certain utilities, toll roads) and that are supervised by public authorities. *Public corporations*, such as cities and certain departments of local and state governments, also provide public services but are financed by the state. Each of these present certain variants on the concept of corporation we have just described.

Additionally, the concepts of state and federal governments themselves present a challenge to ontology. Indeed, they appear to be corporate-like entities, having no essential physical existence. However, in these cases one cannot appeal to a larger deontic framework as the basis for their definition, for they *are* this framework. Instead, at least in democratic societies, one would appeal to the consensus of the voting population (present and past) as a deontic basis. However, since the objectives here are primarily concerned with commercial and financial activities, discussion is confined only to the three classes of economic actors described above: proprietorships, partnerships, and private corporations. Hence,

$$\text{ECON-ACTOR}(x) ::= \text{PROPRIETORSHIP}(x) \vee \text{PARTNERSHIP}(x) \vee \text{PRIVATE-CORPORATION}(x).$$

F. ECONOMIC OBJECTS

Physical Objects

The most obvious type of economic objects are physical (having mass). As before, to individuate these types of entities we usually locate them in the spatial/temporal framework. For most types of physical objects we think of—e.g., tables, chairs, automobiles, real estate, this is unproblematic. However, when granular substances such as corn and wheat, or liquids or gases are involved, problems of identification arise because of the fluid movement of these materials. For instance consider a contract to buy a certain volume of ocean water located at a certain latitude and longtitude at a given depth, etc. Though the geographical coordinates may be certain, the particular volume of ocean water at this location is not.

The practical device that resolves this logical problem in nearly any reasonable commercial context is that of a *container*. Liquids, gases and grains are always handled in a container of some sort, and the container provides the fluid substance with a unique and stable spatial/temporal location and with that discrete identifiability.

Thus, our attention here is confined to what we call *discrete-physical- objects*, which have distinct spatial/temporal coordinates (for instance at their center of gravity) and can be uniquely identified and named. Liquids, gases and grains are assumed always to appear within discrete containers so that the filled container is itself a discrete physical object.

Accounting is however only concerned with those types of objects that can be owned. Normally, any discrete physical object can be owned; however most current legal systems specifically exclude one type, persons (slavery having been abolished). Hence, we introduce a concept of LPHYS-OBJ (legal physical object), which are those that can be owned:

LPHYS-OBJ(x) ::= DISCRETE-PHYS-OBJ(x) & ~PERSON(x).

Promissory Objects

If one examines the asset side of the balance sheet of a company (categories of what the company owns) one of course finds a number of categories that are types of physical objects, e.g., land, plant and equipment, inventory. However, beyond these there are typically other categories that do not comprise physical objects—e.g., accounts receivable, negotiable securities, patents, licenses.

These are what we call ***contractual objects***. They arise as the result of a contractual ***permission*** of which the company is the beneficiary, i.e., they are 'rights' permitting the company to do something (as with licenses) or ***obligations*** of other parties to the company (as with accounts receivables, and negotiable securities).

The equity side of the balance sheet (claims on assets) also has a contractual character, e.g. accounts payables, notes, bonds, preferred and common stock all entail contractual obligations. The logical structure of these contractual objects is the subject of the next chapter.

Monetary Objects

Money is obviously an important type of object in the description of commercial and financial phenomena. If we consider money only in the form of 'hard cash,' i.e., coins and bills, money is simply a type of physical object:

$$\text{CASH-MONEY}(x) \rightarrow \text{LPHYS-OBJ}(x).$$

Coins and bills are obviously of a particular national currency and have a face value. Thus for instance in the U.S., predicates indicating common types of bills and coins are

ONE-CENT-COIN(x)
FIVE-CENT-COIN(x)
TEN-CENT-COIN(x)
ONE-DOLLAR-BILL(x)
TEN-DOLLAR-BILL(x)

etc.

However, in commercial transactions, money is seldom handled at this detail level, but rather as sums of money. In this case we add up the face values of the various coins and bills, and convert them to a common currency unit—e.g., cents or dollars.

Thus, suppose that y is a set of coins and bills, $x_1,\ldots,x_n$. Then the monetary value of y, say n, would be given by a measurement function:

$$y = \{x_1,\ldots,x_n\} \;\&\; \text{MONEY-VALUE}(y, \text{Dollar}, \text{US}) = n$$

Note here that the measurement function has a third place indicating the nationality of the currency, for instance to distinguish measurement in U.S. dollars versus Canadian dollars. (Exchange rates between currencies are described as the tabulated face value of one currency exchangeable for a unit tabulated face value in another currency.)

Most of the examples here assume a US environment. As a notational convenience, the following abbreviation for money in U.S. dollars is introduced:

$$\$(y)=n \quad ::= \quad \text{MONEY-VALUE}(y, \text{Dollar}, \text{US})=n$$

This measurement function is for tabulating face values of a sum of currency in a given nationality. Measuring one nation's currency in terms of another with this function would thus evaluate to zero.

So far we have regarded money as a special type of physical object. However, the services provided by lending institutions in most countries have extended this concept of money.

In the U.S., it is quite common that a bank check is given and accepted in lieu of cash money. These checks are made against 'demand deposit' accounts in a bank, which promises to pay the payee named on the check a sum of money whose tabulated value equals the amount specified on the check.

Demand deposits are thus contractual objects, indicating the obligation of the bank to the party named on the check the specified amount of money.

Because checking accounts are used so often, accounting seldom distinguishes this form of money from actual currency (though they are distinguished in the economic calculation of the money supply, viz. M1 is currency only, M2 includes demand deposits).

The similar function of currency and demand deposits leads us to recognize that currency too has a contractual character. Originally it was a promise by the government to deliver a certain amount of gold or silver on demand. However, it was later recognized that there was nothing unique about these substances as a medium of exchange except their scarcity (and non-reproducibility).

Now however currency (US at least) is no longer backed by gold and silver. Scarcity is maintained through limited printing by the federal treasury. Another important aspect of currency is its relative difficulty of reproduction. This is of course essential to its continued scarcity.

Difficulty in reproduction, it turns out, is a desired feature for contractual objects generally. This is contrary to the aspects of information objects, where reproducibility is desirable. This is further developed in chapter [8].

Information Objects

Another type of owned object might be called an *information object*. Informally, an information object is some meaningful arrangement of symbolic patterns on a representational medium, e.g., ink on paper or electronic codes on a magnetic tape or disk.

The concept of information object here corresponds to what Thompson (1981) calls 'ethereal goods'. He makes the excellent observation that what is distinct about this type of object is the technology of its reproduction. Thus, to him, an ethereal good is one that can be reproduced more cheaply than it can be purchased.

Thus, up until the time of the photocopy machine, a book was not an ethereal good. Now there are certain books that are cheaper to photocopy than purchase from the publisher (especially low volume technical books).

Similarly, home stereo tape recorders made it cheaper to copy musical recordings than buy them.

However, the innovation that really expanded the class of ethereal goods was the electronic computer. A fundamental concept in this technology is that data is easily and instantly copyable. Hence any information converted for computer storage (or indeed programs directing the processing of data) can be instantaneously reproduced (copied to another magnetic medium or sent over communication lines) at practically no cost.

Since considerable labor is often expended in the original creation of such information objects, the legal problem this presents is how to protect the developer from having his/her work "stolen," i.e., reproduced, without compensation.

The use of such terms as 'information object' or 'ethereal goods' may make them sound unnecessarily mysterious. These are only physical objects (media) whose structure or pattern is easily reproduced on other physical objects. Before the printing press, books were less ephemeral in that they had to be manually rewritten. If automobiles could be cheaply reproduced, they too would be ephemeral.

In owning such an information object, therefore, one of course owns the physical representation medium, but more importantly, one owns rights controlling the reproduction of the object. (Thus, the copyright laws for textual material prescribe the "copy rights" of the author and publisher.)

Thus, the important features of an information object for accounting purposes are similar to that of a license, i.e., a contractual permission from one party to another. In the case of information objects, the permitted action is a certain limited range of reproduction. In acquiring an information object, one therefore acquires a physical representation of the information object plus certain rights of limited reproduction.

G. AN OBJECT ORIENTED BALANCE SHEET

From this ontology of economic objects we may sketch a view of accounting prior to the application of monetary valuation. To illustrate, consider the form of a balance sheet containing direct references to objects, as opposed to indirect references in the form of monetary valuations.

On the assets side are various classes of economic objects. These are connected to the accounting entity by means of the OWN predicate.

As a convenience, the graphical notation of Chen (1976) is used to illustrate. Here entity types are drawn as a box, while relationships are shown as a diamond (see Figure [6.1]). (Note that entity types correspond to one place predicates, while relationships are n-place predicates. Arcs correspond to variables in a conjunctive expression.) So for instance an accounting entity, A, might OWN cash, inventory, equipment, etc. (Figure [6.2]).

What is interesting is the role of contractual objects. A contract is between two parties. Accounting only recognizes contracts that are partially executed, i.e., where either additional action is due from the other party, or the firm being accounted. In the first case these are classified as assets, where the OWN relationship applies. In the second case, they are classified as liabilities. We will call the relationship in this case,

DUE(x,y)

where x is an economic actor and y is a contractual object. The balance sheet of organization A then takes the form as shown in Figure [6.3].

Figure 6.1.

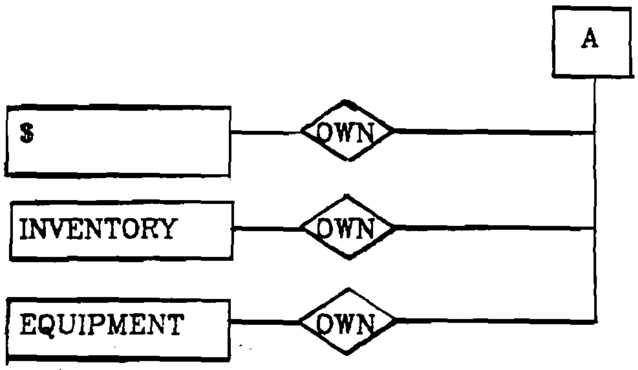


Figure 6.2.

But here we note that an accounts payable to A is an accounts receivable to some other firm, say B. Likewise the bank note is an asset to the bank, the bond is an asset to the bond holder, etc. Thus the broader economy has the form of a chain of contractual objects, as illustrated in Figure 6.4.

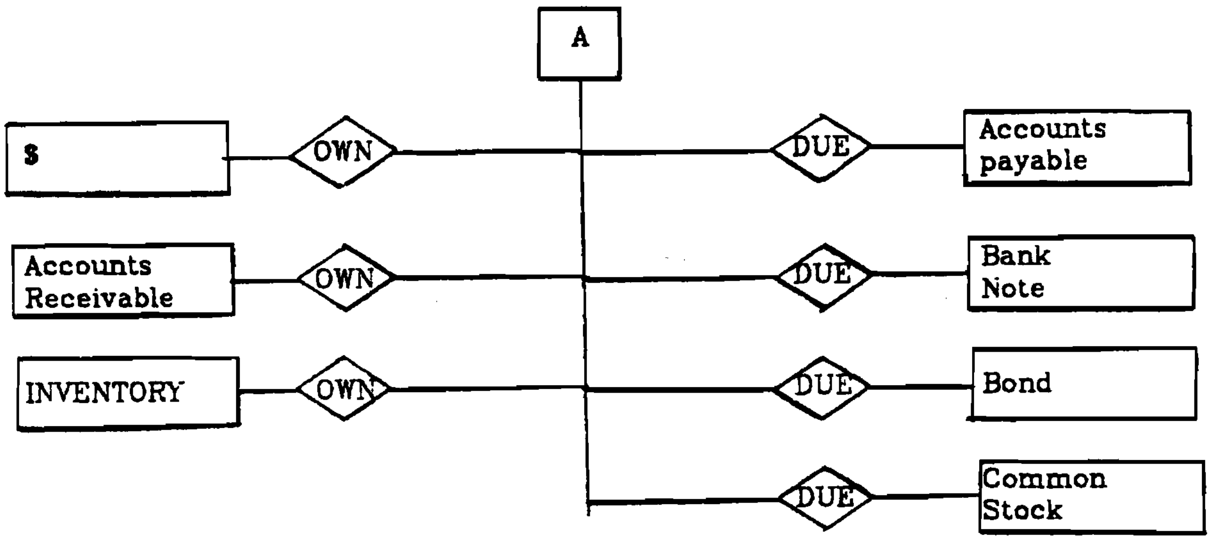


Figure 6.3.

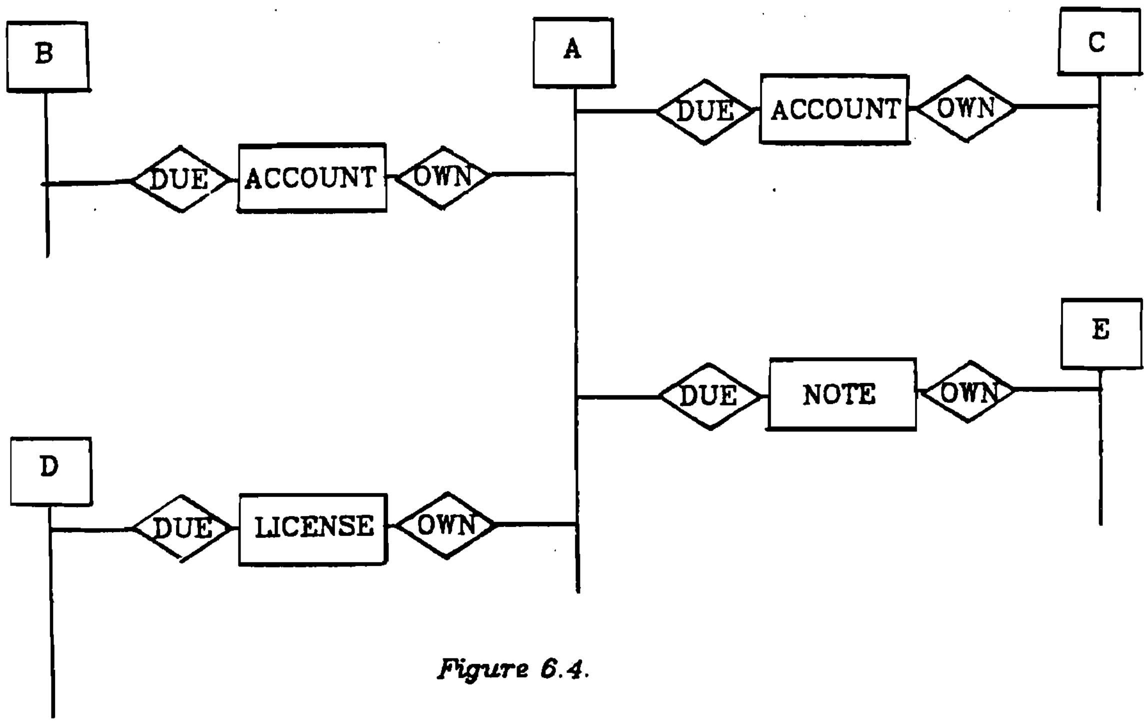


Figure 6.4.

CHAPTER 7:

THE LOGICAL STRUCTURE OF CONTRACTS

CONTENTS

A. INTRODUCTION

In the last chapter, the fundamental role of contractual relationships was observed in accounting theory. Contracts are also of increasing interest in the economic theory of the firm (e.g. Williamson, 1973). In this chapter we explore the underlying logical characteristics of contractual commitment. This involves a branch of logic called ***deontic logic***.

In this chapter an overview of deontic logic is given, evaluating its relevance for inference based decision support systems. Application domains would include interpretation of accounting data and the management of the firm's financial and commercial contracts. In the next chapter, deontic logic is also observed to underly the structure of bureaucratic rules and regulations. Decision aiding systems to interpret complex bureaucratic rule systems as well as to manage their modification will also be a suggested application.

B. DEONTIC LOGIC

1. The Standard System

Deontic logic has its origin in the classical philosophy of ethics. The modern development of deontic logic was initiated in the early 1950's by G. H. von Wright who coined the term, based on the Greek δεόντως meaning 'as it should be' or 'duly'. Deontic logic is a logic of normative concepts. Its major application, outside of ethics, has been to the philosophy of law. It is here that the connections to contract law, and eventually to bureaucratic regulation, might be made.

The basic structure of deontic logic is given by the three deontic operators proposed by von Wright (1951). If q is some arbitrary type of action, then:

O q	means q is obligatory
P q	means q is permitted
F q	means q is forbidden

The sense of these operators obviously relies on what is meant by an 'action'. Von Wright (1968:16) comments:

> A few words should be said about the reading of the formulae. In my first construction of a system of deontic logic the variables were treated as schematic names of actions. ... According to this conception, "Pp" could be read "It is permitted to p". This conception, however, is connected with difficulties and inconveniences. It is, first of all, not clear whether the use of truth-connectives for forming compound names of actions is logically legitimate. It is, furthermore, obvious that, on this view of the variables, higher order expressions become senseless. "Pp" itself cannot be the name of an action; therefore it cannot occur within the scope of another deontic operator either.
>
> It now seems to be better to treat the variables as schematic sentences which express propositions. This agrees with the course taken by most subsequent authors on deontic logic.

> Instead of "proposition" we can also say "possible state of affairs". According to this conception, "Pp" may be read "it is permitted that (it is the case that) p".
>
> Against this reading, however, it may be objected that it does not accord very well with ordinary usage. Only seldom do we say of a state of affairs that *it* is permitted, obligatory, or forbidden. Usually we say this of actions. But it is plausible to think that, when an action is permitted, et., then a certain state of affairs is, in a 'secondary' sense, permitted, etc. too. This is the state which, in a technical sense ... can be called the *result* of the action in question.
>
> We can take account of this combination of action and resulting state of affairs in our reading deontic formulae. Instead of saying simply "to p" or "that p" we employ the phrase "see to it that p". The formula "Pp" is thus read "it is permitted to see to it that (it is the case that) p" or "one may see to it that p". It should be noted, however, that this reading, though convenient and natural, is somewhat restrictive since it applies *only* to norms which are rules of action.

The above three operators reduce to the single operator, O, through the following definitions:

$$Pq \leftrightarrow \sim O \sim q$$

reading: that q is permitted means it is not obligatory not to q.

$$Fq \leftrightarrow O \sim q$$

reading: q is forbidden means it is obligatory not to q.

Discussion typically focuses on the interplay between obligation and permission. (However, in legal and bureaucratic contexts, prohibition – i.e. forbidding – has an important background role. We return to this later.)

The practical relevance of deontic logic in administrative contexts is to provide automatic inference in, say, contract arbitration or the interpretation of bureaucratic regulation. Such applications are useful in complex cases where the chain of connections would otherwise be difficult to follow. Thus the axioms and inference rules of deontic logic take on pragmatic importance that the system draws the correct and intended conclusions.

Various axiomatic systems have been proposed. In an introductory survey, Føllesdal and Hilpinen (1971) present what they call the 'standard' system of deontic logic. Based on propositional logic this serves as a more or less consensually accepted core on which to base further discussion. The standard system assumes elementary generic actions (in the sense of von Wright, above). Assuming p and q to be actions of this type, the standard system begins with the earlier definition:

(D1) $$O\,p \leftrightarrow {\sim}P{\sim}p.$$

Three axioms follow:

(A1) $$P\,p \vee P{\sim}p.$$

This is the 'principle of permission': for any act p, either p or ~ p is permitted.

(A2) $$P\,(p \vee q) \leftrightarrow P\,p \vee P\,q$$

This is the 'principle of deontic distribution': that p or q is permitted if and only if p is permitted or q is permitted.

(A3) $\sim P\ (p\ \&\ \sim p)$.

This axiom, not included in von Wright's original formulation, says that it is not the case that both p and ~ p are permitted. Using definition D1, these axioms can be re-stated in terms of obligation:

(A1') $O\,p \rightarrow \sim O \sim p$

reading: if p is obligatory, then it is not obligatory not to p.

(A2') $O\,(p\ \&\ q) \leftrightarrow O\,p\ \&\ O\,q$

reading: if p and q are together obligatory, then, separately, p is obligatory and q is obligatory.

(A3') $O\,(p \vee \sim p)$

reading: either p or not p is obligatory.

Added to this system (either set of axioms) are the inference rules of propositional logic (substitution, modus ponens) plus the rule:

(R1) If 'p' and 'q' are logically equivalent, then 'Pp' and 'Pq' are logically equivalent.

Here, p and q are regarded as propositional variables. However, they are not exactly propositions in the usual sense of referring to a static state of affairs, e.g. the window is closed. Rather, as names for generic actions, they refer to someone's causing a certain state to occur, e.g., closing the window. Thus, in this form of deontic logic, the concept of truth value is replaced by one of performance value, i.e. whereas a proposition is either true or false, an action is either performed or not performed. Further, the actions controlled by these deontic operators presume an aspect of human agency. We don't obligate or permit natural phenomena such as the sun rising.

The propositional form of deontic logic could be refined to distinguish individual agents and the objects they act upon. This is in particular necessary for applications to contractual relationships and bureaucratic regulation. These refinements, and the complications they entail, will be considered later. At the moment however the concern is with the interactions and interpretation of the deontic operators themselves, and so we remain with an undecomposed concept of action.

2. Deontic Paradoxes

Computer applications of deontic logic to organizational and economic contexts would be most useful in cases where a complex system of rules and regulations was involved. Here, the machine would assist in tracing through the various implications of a contract or the regulations pertinent to a proposed action. In such cases, where we begin to rely on the machine to follow a long deductive maze that we ourselves have difficulty following, it is vital that we have an absolutely solid confidence in the types of deductions that the machine makes (as we have for instance in the arithmetic deductions made by machines).

In this regard, various apparent paradoxes that arise from the standard system of deontic logic are disturbing and require attention. Well known among these is one first noted by Ross (1941), hence known as Ross's paradox. It centers on the interpretation of the theorem:

(a.) $$O\,p \rightarrow O\,(p \vee q)$$

which can be derived from the above axioms. This says that if a certain action p is obligatory, then p or q is obligatory. Thus, if I ought to mail a letter, than I also ought to mail or burn it.

A related paradox, based on permission, is given by the theorem:

(b.) $$P\,p \rightarrow P\,(p \vee q)$$

which says that if p is permitted, so too is p or q. Hence, if I am permitted to smoke, I am also permitted to smoke and kill. Føllesdal and Hilpinen (1971) observe that these are problems due not to the deontic operators, but to the interpretation of the propositional connectives, 'V' and '→'.

Logic, in formalizing ordinary reasoning processes, tends to draw from terms in ordinary language, but use them in more specialized ways. In the case of the propositional connectives, their usage is rigorously determined by truth tables, whereas in natural language, the corresponding terms 'and', 'or' 'implies', etc. is less rigid. Thus in logic, the statement:

I walk to work V I carry my lunch

follows from

I walk to work

This seems anomalous because the logical disjunction, V, is inclusive whereas the ordinary English 'or' is typically used exclusively. Føllesdal and Hilpinen explain:

The paradoxes mentioned above may perhaps be explained by reference to very general conventions regarding the use of language. For instance, it is generally assumed that a person makes as strong statements as he is in a position to make. If someone wants another person to mail a letter, it is surely very odd for him to say that the letter ought to be mailed or burned, especially if the latter alternative is forbidden. Similar remarks apply to [b. above]. If we want to explain the actual uses of deontic expressions in ordinary language, such general conventions must be taken into account, but they need not be incorporated into deontic logic.

In the case of the 'paradox' involved in [b. above], it is important to observe that in ordinary language, the logical force of the word 'or' is in some cases the same as that of 'and'. For instance, in many cases the sentence 'a may do p or q' is used to express the same statement as 'a may do p and a may do q'. This fact has led some philosophers to assume that these cases involve a special notion of permission, termed ***free choice permission***. G. H. von Wright (1968) has suggested that a free choice permission and the permission concept defined by the standard system of deontic logic have different logics; the former concept does not satisfy the distribution principle [A2], but instead the law

$$P(p \vee q) \leftrightarrow Pp \,\&\, Pq$$

The 'paradoxical' theorem [b.] is not valid for this notion of permission. According to von Wright, the notion of free choice permission cannot be formalized in the standard system. It seems to us, however, that a free choice permission *can* be expressed in the standard system in a perfectly adequate way: P p & P q. If 'a is permitted to smoke or kill' is a free choice permission, it should be formalized as P p & P q, and this is not, of course, implied by P p (according to the standard system). If the word 'or' is interpreted in this way (as it often is in ordinary language), 'a is permitted to smoke' does *not* imply 'a is permitted to smoke or kill'. There is no need to invent special notions of permission or construct special logics of permission and obligation on the basis of this accidental interchangeability of the words 'or' and 'and' in ordinary language.

For a more extensive discussion of deontic paradoxes arising from varying interpretations in natural language, see Casteñeda (1981).

3. The Semantics of Deontic Operators

The remarks in the preceding section about the differing interpretations of the logical connectives and their natural language counterparts leads to similar issues in the interpretation of the deontic operators themselves. This has pragmatic importance in computational applications for, even if the system's deontic deductions are logically correct, they may nonetheless be mis-interpreted by users of the system*.

On this point, Føllesdal and Hilpinen (1971:15) observe:

> Deontic formulae are normally interpreted simply by translating them to sentences of ordinary language. The plausibility of putative theorems is judged on the basis of the intuitive plausibility of their ordinary-language counterparts. ...
>
> The formulation of our intuitions concerning deontic notions in ordinary language often involves ambiguous expressions such as 'implies', 'requires', etc. In many cases it is difficult to see what are the exact formal counterparts of these intuitions, that is, what our intuitions really pertain to. ... Moreover, a 'literal' translation of formulae such as $O\ p \rightarrow O\ O\ p$, $P\ O\ p$, etc., to ordinary language yields sentences which are hardly ever used at all. It is almost impossible to decide whether such sentences are acceptable as principles of deontic logic or not.

Formal semantical aspects are therefore especially important in the case of deontic logics. Useful insights are provided by von Wright (1968). Here, deontic logic is viewed as a branch of modal logic. In model logic, the central issue is an explication of contingent vs absolute truth of assertions. The two basic concepts are therefore *possibility* and *necessity*. For u an arbitrary assertion, these are typically denoted as:

* An application scenario might be that the machine maintains an internal representation of contracts or bureaucratic regulations in some extended form of a deontic logic. In responding to user queries, these are translated to an appropriate English form by means of a text generating grammar. Automatic text generation from formalized representations is discussed in Lee and Krigman (1978) and McDonald (1977).

$\Box$ u (u is necessarily true)

$\Diamond$ u (u is possibly true)

As for permission and obligation, possibility and necessity are inter-definable:

$$\Diamond u \leftrightarrow {\sim}\Box{\sim} u.$$

That is, possibly u is equivalent to not necessarily not u.

Kripke (1963) provided a formal semantics for modal logics based on possible worlds. Necessarily u meant that u was true in *all* possible worlds. Possibly u meant that u was true in *some* possible world.

Semantically, von Wright viewed the deontic operators as qualifications of these modal concepts. Permission is regarded as deontic possibility, obligation is deontic necessity. The extension of these concepts ranges over a subset of all possible worlds, namely those that are 'deontically perfect', conforming to the normative (legal, ethical) system under study. That is to say, if q is obligatory, (O q), the q will be true in *all* deontically perfect worlds. If q is permitted, then q will be true in *some* deontically perfect world. Note however that since O q does not imply q (not all obligations are fulfilled), the actual world may not be included in the set of deontically perfect worlds.

Recalling the discussions in chapters [3] and [4], it may again seem that all this talk of possible worlds is getting too mystical sounding if we eventually intend to get down to such mundane realities as contracts and bureaucratic regulations. We don't expect that sales contracts or income tax forms will ever contain clauses referring to deontically perfect worlds.

However, the elaboration of the logics underlying these documents may depend on them.

C. TIME, CHANGE AND ACTION

As noted, the deontic operators apply to generic actions that are regarded as propositions. However these are not propositions in the usual sense of describing some static state of affairs. Rather, actions involve two additional components:

a) a change of state

b) due to some human intervention

As these are non-trivial differences, actions themselves have been the object of philosophical study, resulting in proposals for various 'logics of action'.

It is typical to separate the two aspects and regard the logic of action as building upon a logic of change. Since change is a temporal concept, this relates to concerns of temporal logic.

1. Temporal Logic

Propositional logic normally involves statements like:

snow is white

In these cases, the truth of the sentence does not depend on time, i.e., the implicit claim is that snow is always white in this and any possible world. However, statements like

it is raining

have a more restricted claim, i.e. that it is raining now, at some specific time and place. Rescher and Urquhart (1971) for example make this temporal aspect explicit with an operator R for 'realization'. Thus a proposition

$$R(t)\ p$$

has the reading that p 'is realized' at time t. Assuming time to be a linear dimension along which propositions are true over certain intervals, false over others, this is relatively straightforward.*

Rescher and Urquhart in fact note the parallel to a positional logic indicating the geographical dependence of certain propositions. For instance, analogous to the R operator for temporal realization, another operator R' might be defined,

$$R'(l)\ p$$

with the reading that proposition p is realized at location p. For instance, the proposition 'there is oil here' is true of some places, not of others.

Realization operators of this sort do not however differ markedly from ordinary predicates. For instance if the ontology includes places, and/or times

HAS-OIL(l)

RAINING(l,t)

might be used to express that location l has oil or that it is raining at location l at time t. Distinguishing places and times from other types of individuals for separate treatment seems to be little more than syntactic

* A complicating factor is whether time is viewed as discrete units or a continuum. In this section a discrete view of time is adopted. In Lee (1980a, 1981), a continuous time dimension is assumed.

engineering. The motivation would be if the aspects of location or time are sufficiently subtle and complex to warrant separate study and axiomatization. In this regard time does seem to have epistemological prominence.

In chapter [4], situating an object in the spatial/temporal framework was observed to be a useful basis for consensual individuation. We have strong social agreement on the typology of space and time. These are not the scientific concepts of space held by the astronomer, nor the view of time of the physicist, but are more mundane. What matters is not what space and time 'really' are, but how we think about them, e.g. in administrative transactions.

For these purposes, the world is basically two dimensional. We think normally about geographical location using east-west and north-south dimensions and only on special occasions, e.g. tall buildings, airplane landings, do we worry about the vertical dimension. Moreover, for the purposes of ordinary discourse, the world is still flat. We may know scientifically that the planet is spherical, but in our transactions we treat it as a plane.

Time is often regarded as simply another dimension to this geographical framework. It is however a dimension with a special epistemological status for management. Business transactions for instance are careful to note the identity of the buyer and seller but their geographical location at the time of the sale is usually of minor importance. As telecommunications increase (leading to such concepts as 'telework') geographical position becomes even less consequential. On the other hand, temporal relationships are extremely important in all types of economic activity. In accounting, every transaction is dated. The accounting conception of cost is time dependent. Contracts specify due dates and time dependent penalty clauses. Planning, often considered to be the most important of management functions, is a temporal concept. Time (our ordinary conception of it) differs from the geographical dimensions also in that it is ***ordered***. Time moments are 'later than', 'earlier than' one another. This, in itself, is no great difficulty. The complications arise in that we seem to maintain two perspectives of time, one historical, the other looking towards the future.

Our perception in these cases is quite different. As we view the past, we normally regard time as a single line along which historical events are ordered. It is relatively easy for us to decide whether two events coincided, whether two authors were contemporary, etc. Our projection of this time line into the future doesn't have this same neat arrangement of events. We don't know for instance where to temporally locate the company's next technical innovation or the next major sale.

But here it is not that our conception of the time dimension is different, but rather that we lack the knowledge of future events that we have for historical events. As discussed in chapter [4], this conception of time is regarded as 'backwards linear and forward branching' (e.g. Rescher and Urquhart, 1971, McDermott, 1982) as in Figure [7.1]. The horizontal scaling of this dimension in this view is uniform. For example, the Gregorian calendar assigns arbitrarily distant dates for the past and the future. The graphical branching reflects contingencies in future events. An analogous view is reflected in such planning tools as PERT charts and decision trees.

Let us return now to the remarks concerning the notation for time, e.g. in a predicate language.

2. A Logic of Change

In the view presented above, using either a realization operator, or providing predicate places for time variables, we are able to express that certain prospectives are true of the world only at certain times. The emphasis is on when certain states obtain.

By contrast, a proposal by von Wright (1965) focuses more directly on *changes* between states. He introduces a new propositional connective, T, read 'and next'. If p and q are propositions describing generic states of affairs,

p T q

is read "p and next q", i.e. that q is the state following from q. For example,

raining T sunshine

expresses that it has stopped raining. Note that the arguments are generic states of affairs, i.e. not bound to a specific time. The expression therefore describes a generic change. Considering only a single proposition, q, four elementary changes can be distinguished:

~q T q (q begins)

q T ~q (q ends)

q T q (q continues)

~q T ~q (not q continues)

It may be debated whether the last two are changes in the usual sense. In any case, these prove to be useful distinctions in the explication of action and responsibility.

3. A Logic of Action

Actions differ from simple changes in that they contain an additional component of human responsibility. Actions are changes that people bring about, such as closing a door as opposed to the wind blowing it shut. The implication is that the resulting state of affairs would not have occurred without the person's interference. Actions therefore contain a hypothetical aspect: what *would* have been the case without the person's intervention.

Recognizing actions therefore involves not only observation, but a certain amount of theorizing. The sun rising is not a human act like closing the door because our theories tell us that the sun will rise independently of our behavior. A dancing Indian might be seen by some people as causing the ensuing rainstorm while others, with different theories, will claim it to be a mere coincidence of events. Included here is the supposition that the person's behavior *causes* the event. This might be either direct or indirect causality, e.g. a shop manager may cause a production batch to be created by giving verbal commands.

Causality has however been a long standing philosophical thorn. Hume (1739) noted that causality can never be proven through empirical observation alone. Causal statements are therefore theories not only about actual states of the world but possible states as well.

These aspects are incorporated into the concept of action developed by von Wright (1967). Extending his previously mentioned concept of change, he adds another propositional connective, I, read 'instead of'. This combines with the T connective to form 'TI expressions' describing actions. If P, q and r are propositions for generic states of affairs, a TI expression is of the form,

$$p \; T \; (q \; I \; r)$$

read that 'p and next q instead of r'. The interpretation is that p and q are the observed states, before and after, of the action. The proposition r describes the state that would have obtained without the agent's interference. As might be expected, the formal semantics of these expressions includes possible worlds. The propositions p and q are true of the actual world at successive times, say t_0 and t_1. The proposition r describes an alternative possible world, at time t_1, which otherwise would have been actual (see Figure [7.2]). The T connective relates two states in the actual world at two successive times, whereas the I connective relates alternative possible worlds at the same time.

D. CONTRACTUAL COMMITMENT

We began with the standard system of deontic logic having the operators:

O q	q is obligatory
P q	q is permitted
F q	q is forbidden

In the original interpretation, q represented a proposition, i.e. a state description. Under von Wright's interpretation, the variable q is replaced by TI expressions, whose internal structure has three state descriptions: the current state, the next state if the action is performed, and the alternative next state if the action is not performed. An obligation, for instance, would therefore have the form:

$$O\ (p\ T\ (q\ I\ r))$$

read as: it is obligatory to change the state p to the state q instead of the state r.

We would now like to sketch the extension of these concepts to represent contractual commitment. The discussion here is informal. A more rigorous treatment is provided in the appendix.

In the preceding discussion, the agent of the actions was implicit. To specify the agent explicitly, we would need to add an extra place to the I connective, e.g.,

$$(p\ \ Ix\ \ q)$$

where x is a variable for human individuals. Thus a statement of the form

O (p T (q Ix r))

means that it is obligatory for x to change the world from p to q instead of r. This is a general concept of obligation, e.g. of legal codes or ethical systems.

Contractual obligation, by contrast, is an obligation from one party to another. This requires that we mark the deontic operators with two places indicating the parties to the contracts. For instance,

(x O y) (p T (q Ix r))

would mean that x is obliged to y to see to it that the state p is changed to the state q instead of the state r. We note that with obligation, the party that is obligated is also, typically, the agent of the action. (In the case of contracts between organizations, the agent of the action might also be a subordinate of the person obligated in the contract.) With permission, the matching of arguments is reversed, e.g.

(x P y) (p T (q Iy r))

that is, x permits y to change p to q instead of r. Here it is y who is the agent of the action. (Prohibition, being the simple negation of permission, has a similar syntax.)

Further refinements are needed. Contractual obligation and permission usually have a time period stipulated. Using the aforementioned R operator, we might denote this as:

$$(x\ O\ y)\ R(t)\ (p\ T\ (q\ Ix\ y))$$

Here t is meant to indicate a time interval. That is, x is obligated to do the prescribed action within time t. Similar time limitations might be applied to permissions.

What we have so far described is however only half of what we normally regard as a contract. That is, a contract normally entails a concept of exchange, e.g. of money for goods, services or privileges. Abbreviating an action done by x and y as p(x) and q(y), the elaboration of a typical sales contract would therefore look like:

$$(x\ O\ y)\ p(x)\ \&\ (y\ O\ x)\ q(y)$$

i.e., x is obligated to y to do p (e.g. deliver goods) and y is obligated to x to do q (e.g. pay a certain amount of money). Often, there is a certain inter-dependence between the two actions, for instance, y's obligation to pay may only apply after x has delivered the goods.

This introduces the important aspect of *contingent* obligation (permission, prohibition). Other examples of contingent aspects are penalty clauses and insurance contracts. Indeed, the failure to execute a contractual obligation is normally governed by the more general legal obligations of commercial law. The injured party therefore has the privilege (permission) to initiate legal action under these circumstances. Contractual obligations therefore interface with general legal obligations in their enforcement.

While contingency is a fundamentally important aspect of contractual commitment, it is – unhappily for application purposes – still an unresolved area within deontic logic. Von Wright(1968) proposes a dyadic or conditional counterpart to the standard deontic system, where

$$O\ (p\ /\ q)$$

is read that p is obligatory *given that* q. This proposal has not been widely accepted, even by von Wright himself. In von Wright(1971:160-1), he argues that contingency requires a new conception of deontic logic, "not immediately as an ***analogue*** of modal logic, but as a ***fragment*** of a more comprehensive logical theory [to be called] the Logic of (Sufficient and Necessary) Conditions".

A more recent proposal, by Thomason (1981), argues that the problems presented by contingent obligation are resolvable through recognition of temporal aspects. He comments (1981:165-166): "Most of the recent work in deontic logic has concentrated on problems concerning 'conditional obligation' ... I want to claim that deontic logic requires a foundation in tense logic; the notion of obligation is so dependent on temporal considerations that a logical theory of obligation pre-supposes an appropriate logical theory of tense."

The argument is that contingency, in deontic contexts, typically implies temporal ordering. If I promise to give you an apple if you wash the dishes, my obligation is typically understood to begin *after* the dishwashing. A more detailed discussion of these aspects is provided in the appendix.

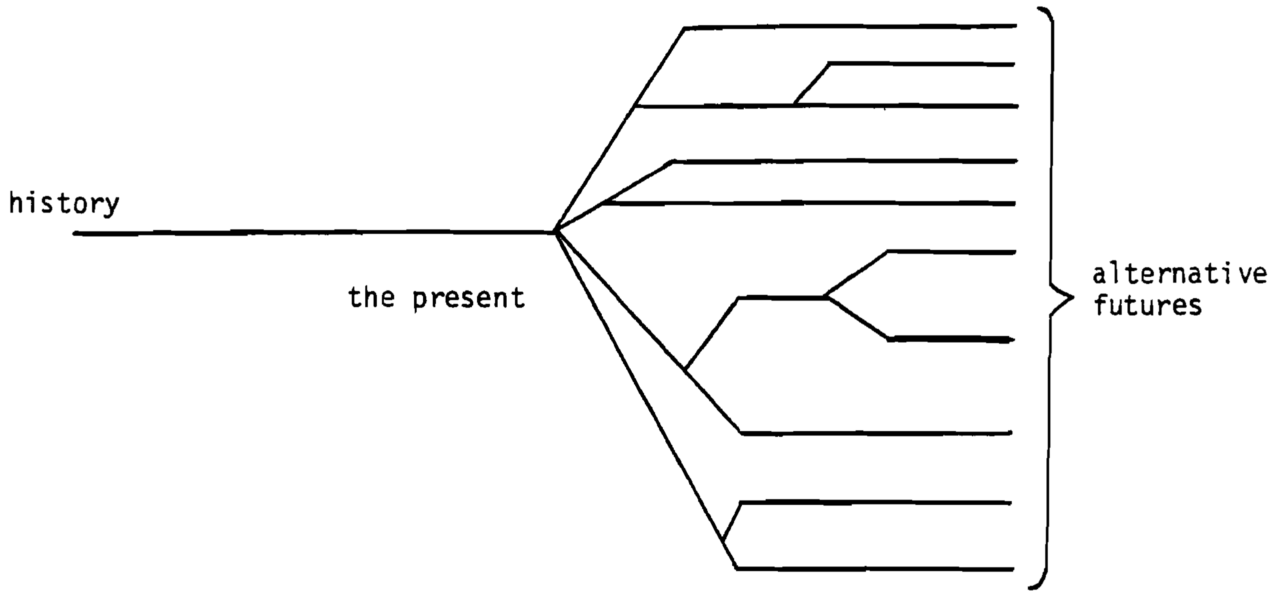


Figure [7.1] Backwards Linear, Forward Branching View of Time

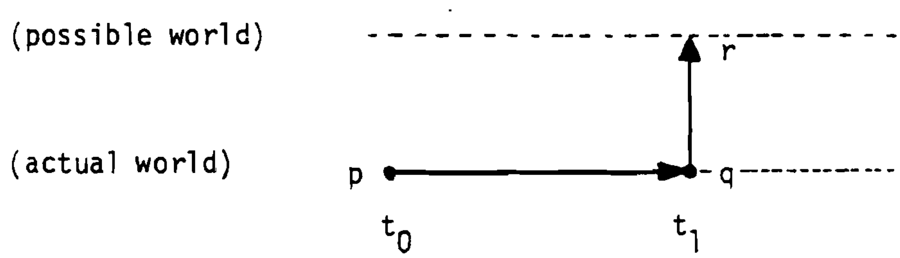


Figure [7.2] Von Wright's Concept of Action

CHAPTER 8:

ANALYZING RED TAPE: DEONTIC PERFORMATIVES

CONTENTS

A. INTRODUCTION

> ***red tape*** n {so called from the red tape formerly used to tie up legal documents in England}: bureaucratic procedure, especially as characterized by mechanical adherence to regulations, needless duplication of records, and the compilation of an excessive amount of extraneous information resulting in prolonged delay or inaction. (Webster's 3rd International Dictionary).

Red tape is an irritation that most people accept with a certain amount of fatalism, like catching colds in winter. Just as medical science has had little impact on the common cold, management science and information technology seem to have had little effect on reducing bureaucratic red tape. It seems to be a natural by-product of organizational and societal rationalization.

Office automation, however, would seem to have as an implicit goal the reduction of red tape. Part of this is (rightly) seen as the elimination of paper flows. Documents can be handled much more quickly and efficiently in electronic form than as physical paper. But there is another component to red tape, a sociological one, which tends to be ignored. Red tape arises as authority structures become specialized and distributed across numerous organizational roles.

Much of what we call red tape involves the processing of a particular request through a series of authority nodes (typically offices) in the organization. Thus another part of the problem, beyond speed of communications, is the resource time at these nodes —i.e., the time taken by the particular clerk or manager to authorize the request. Still another part is finding the appropriate authorities in the first place. (Another piece of informal terminology applies here: 'passing the buck.')

The basic point is that the problem of red tape involves not merely information flows but also authority flows. What is meant is not the broad types of authority typically drawn on organization charts, but rather the detailed, formalized types of authority prescribed in bureaucratic rules and regulations. An important aspect is that often these types of authority have also come to be ritualized, that is, no longer relevant to the organization's interests.

Authority is of course a sociological phenomenon. That is not to say it is not analyzable. The more specific point of this paper is to sketch an approach to the analysis of red tape.

The approach is introduced through a linguistic distinction between performative vs informative documents. These are regarded as the basic medium of bureaucratic authority. These are generally recognized by the inclusion of a signature by the authorizing person or a special stamp or seal of the authorizing office. The sociological importance of the non-duplicatability of these documents is discussed.

The content of authoritative documents is analyzed using the primitive operators of deontic logic (obligation, permission, prohibition). The relationship of these distinctions to a broader theory of bureaucracy is examined, and a concept of bureaucratic software is suggested.

B. DOING THINGS WITH WORDS

The linguistic concept of a performative was first introduced by Austin (1962) and elaborated by Searle (1969) and others. The performative aspects of contracts and financial instruments was discussed in Lee (1980, 1981). The relevance of performaives to office processes was first noted by Flores and Ludlow (1981).

A performative is an utterance that not only conveys information but also, by its being spoken, accomplishes some socially significant act. For instance, the sentence "I now pronounce you man and wife" when spoken by a priest during a marriage ceremony not only describes the relationship between the couple, but actually *creates* it. This example brings out several key features of performatives. One is that the state created by such an utterance is generally some type of social artifact. Obviously, the mere speaking of a few words has very little physical effect. Rather, it places one or more people in different states of social perception. Often, this involves a certain set of obligations, e.g., of fidelity, economic responsibility.

The roles involved in a linguistic utterance are usually cast as speaker and listener. However, in the case of performatives, the listener role must be divided between 'addressees' and 'by-standers'. Clearly, not everybody attending the marriage ceremony becomes socially obligated by the priest's pronouncement, only the two people specifically addressed.

Also, it is not always the addressees of performatives who acquire the social obligation by the utterance. For instance, a major class of performatives is the class of *promises*, in which case it is the speaker who acquires the obligation. In other cases the addressee may in fact be an object, e.g., a ship: "I christen thee the Queen Elizabeth." These latter are, however, fairly rare types of performatives.

The social contract surrounding a performative is not always institutional, as with marriage. For instance, such remarks as "I promise to do the dishes tomorrow," are also performatives. Here, however, attention is limited to performatives in institutional environments. In these cases the speaker and addressee must have certain social qualifications in order for the performative to have force. For example, only priests, ministers, ship captains, justices of the peace, etc., can pronounce marriages, and only unmarried couples of a certain age can become married. Further, apart from the broad social context that enables the performative to have force, for instance the church as an institution, there is also a narrower, 'conversational' context where the performative must appear. For example, the marriage pronouncement must appear at a certain point near the end of the marriage ceremony, not at the beginning, nor afterwards, during the reception, etc.

C. WRITTEN PERFORMATIVES

Linguists generally refer to performatives as a type of *utterance*, that is, a spoken communication. What is sometimes overlooked is that written communications, too, may be performative. In these cases, however, the execution of the performative takes on a somewhat different character. In a spoken performative, the person making the performative is obviously identified as the speaker. In written performatives, the issue of authorship arises. Also, with spoken performatives the addressee hears the performative at the time it is spoken. Written communications, however, endure throughout time and so the addressee may receive the communication considerably later than when it was initially made. The question then arises: when during this interval does the performative come into force?

These issues of authorship and timing are commonly resolved by a very simple device, namely the author's handwritten signature, accompanied by the date on which it was signed. The ritual of signing one's name to a document is so pervasive that its fundamental role is often not recognized. Indeed, as a rough heuristic, one can usually distinguish purely informative documents from those with a performative component by whether or not it has a personal signature. For instance, printed announcements, bulletins, etc., seldom have signatures; contracts to pay money (checks, etc.) always do. The effect of the signature is roughly the declaration:

> "I hereby acknowledge that my beliefs and intentions are accurately described by this associated text."

Signed documents, as performative instruments, also acquire a unique feature not possessed by their purely informative counterparts: the performative effect of the original signature is not carried over to its mechanical duplicates. For instance, in legal documents, such as contracts, wills, etc., when several copies are made, each must be separately signed by the author(s) to have legal validity.

The unique role of the original in written performatives has, by the way, its counterpart in spoken performatives as well: repeated playbacks of a tape recording of a spoken promise, for instance, do not create new promises. With written performatives the assumption of course is that the signature provides a unique identification of the author. However, the authenticity of the signature is seldom called into question (handwriting analysts are seldom needed in court). A more important effect is that it signals the author's declaration of *personal responsibility* for the associated statements. In the act of signing such a document the signer typically becomes acutely aware of its language and contents (especially if the text has been written by someone else, as in a standardized lease or loan contract), since (s)he is henceforth expected to behave in accordance with this declaration.

The social significance of this ritual, committing the signer to having the beliefs, attitudes or intentions as expressed in the document, has been accepted by nearly every literate culture for centuries. It is an extremely useful historical convention, being the hallmark of honesty and good faith in all kinds of institutional and governmental transactions and agreements. It should be noted, however, that a signature is not the only way of marking a performative document. In many cases, a special seal, stamp or sticker operates similarly, especially where the effect of the document is standardized and commonplace. Typically, these special performative symbols are designed with a special, intricate pattern that would be hard to mimic. Often, these serve effectively as the signature of an institution, rather than a single individual. Common examples are coins, bills, and postage stamps.

D. DEONTIC PERFORMATIVES

In the context of organizational procedures, the informative/performative distinction can be refined further. One aspect of these procedures is certainly to transmit and store information. Another, however, is to control and standardize the behavior of the personnel involved. Procedures are thus means of standardizing the exercises of *authority* of certain individuals in the organization over others.

Authority, of course, includes a wide variety of aspects. With regard to red tape, however, one particular form of authority seems prominent. This is where a certain type of behavior is in general forbidden, except under special circumstances. The exercise of authority in these cases amounts to some person's evaluation of the circumstances, and the granting of *permission* where appropriate. In many instances of red tape the action in question is divided into a number of sub-actions each requiring separate permission. The delay or inaction inherent in the definition of red tape thus results not for reasons of information collection or processing, but rather due to the wait times in the personal queues of these various permission granting individuals.

A familiar example of this is automobile registration. In general it is forbidden to drive an automobile on public roads. There are, however, several conditions that together permit this. First, the driver needs to be able to drive. This is demonstrated by an examination by state employees with the authority to certify driving skills. If the driver succeeds in this exam, the examiner signs the examination form that permits the driver to obtain a specially designed (performative) card, the driver's license.

Next, one must have an automobile. In purchasing the auto, another special form is required — the bill of sale and/or title certificate — which is signed by both the previous and new owners (another performative document). Next, the automobile itself must be in safe driving condition. Here, a different individual, e.g., a state licensed mechanic, makes the certification. This is typically signified by a special (again performative) sticker attached to the auto's windshield or fender, signed by the mechanic. Next, if not already done, the vehicle must be registered, i.e., recorded in the state books. Here, typically, the vehicle manufacturer's serial number is recorded by another state agent on another special form, which (s)he signs. This permits the owner to obtain a license plate for the auto (analogous to a performative seal). Lastly, in some places, a separate road tax must be paid. Here again, receipt of payment is acknowledged by a special receipt form and/or sticker (more performative items).

The sum of all these procedures amounts to permission from the state to drive the vehicle on its public roads. Note that the component performatives in this case were sometimes marked by a signature, sometimes by a special seal or sticker, and sometimes both.

Similar types of permission structures exist within organizations. Here a common example is the request of some department to purchase a large item. Often such a request must be approved by a number of individuals to verify for instance that the item is technically sound, compatible with similar items in the organization, competitively priced, etc. In each step along the way, the permission performance is inevitably signaled by the signature of the authorizing individual.

Another common type of organization performative is order giving. Interestingly, this seems to be a more efficient process than permission granting. The difference seems to be that orders are generally given by a single individual to a number of others, whereas permission often needs to be granted by a number of people together for a single person. For this reason, perhaps, order giving seems less involved in the concept of red tape.

There is, however, an interesting duality between permission granting and order giving. This relies on the discussion of deontic logic from the preceding chapter. Let "q" symbolize some particular type of action. Then the following operators are used:

O q (q is obligatory)

P q (q is permitted)

F q (q is forbidden/prohibited)

Without going into any more logical details, two interesting points can be brought out. The first is that permission and prohibition are negates. That is, to permit some action is not to forbid it and vice versa. Symbolically,

$$P\,q \leftrightarrow {\sim}F\,q$$
$${\sim}P\,q \leftrightarrow F\,q$$

The more interesting insight, however, is that obligation and permission are logical duals. That is, to be obliged to perform some action, q, is equivalent to not being permitted not to do it. Conversely, being permitted to do a certain action is to not be obliged not to do it. Symbolically,

$$O\ q \leftrightarrow {\sim}P{\sim}q$$
$${\sim}O{\sim}q \leftrightarrow P\ q$$

The relevance of this to the discussion at hand is that it suggests a family of what might be called 'deontic performatives' that are inter-definable. A deontic performative document is one that obliges, permits or forbids some action. These are important in that they indicate the link between performative documents and authority structures.

Let x and y indicate two people or roles in the organization. Then the preceding notation can be modified to indicate three basic types of authoritative action:

$$(x\ O\ y)\ q = \text{x orders y to q}$$
$$(x\ P\ y)\ q = \text{x permits y to q}$$
$$(x\ F\ y)\ q = \text{x forbids y to q}$$

The enabling requirement in each of these cases is that x has the *authority* (within the organization) to control y's behavior in doing q. The argument we want to make is that signed, performative documents nearly always signal a change in deontic status.

Lee (1980 and 1981) analyzes the deontic structure of contractual relationships *between* organizations. Indeed, nearly all inter-organizational transactions —with the exception of cash sales —involve a deontic aspect.

For example, credit sales and bank loans, bonds, certain types of preferred stock, etc. create an obligation to a later payment action. Insurance contracts establish a contingent obligation of the insurer to the insuree. Easements and licenses of various kinds establish a permission relationship between the parties.

In each case, the signing of the contract creates a change in deontic status. For example, signing a bank note creates an obligation to pay that previously did not exist. An easement creates a permission to limited use of another's land, altering the general prohibition against trespassing.

Our suggestion here is that a similar view applies to transactions *within* an organization. The red tape within organizations shares many characteristics of contractual relationships between organizations.

E. THE INDIVIDUATION PROBLEM

In the last couple of decades, the analysis of document processing and flows in organizations has become closely coupled with efforts to apply computer based information technology to the task. The most substantial change introduced when a particular document process is automated is that the documents themselves no longer have a fixed physical counterpart as paper, but are instead only magnetic or electronic patterns. This offers enormous flexibility for information transmission and processing; transfer of the document from one geographic location to another is effectively instantaneous. Likewise, several people can simultaneously work on different parts of the document at the same time, since they may all access a centralized representation of it.

While this technology is especially well-suited to handling the *informative* content of documents, it does not accommodate documents having a *performative* aspect. This is due to the fact that in paper form, a performative document has a physical uniqueness that it loses when converted to a magnetic medium. For physical representations, we have clearly developed concepts of individuality and uniqueness. When we move a physical document from one place to another, we know for instance, that it is the *same* document; whereas, if we see two duplicate documents, we know they are not the same since they occupy different physical locations at the same time.

The sameness problem is an old philosophical chestnut. It is often illustrated by the so-called 'ship of Theseus.' Imagine a wooden boat. We replace one plank. Is it the same boat? Now systematically replace all the planks. Is the second boat now the original boat? (A whole navy of the same ship can be built by iterating this process). Clearly, where we draw the line between the original boat and its duplicates is a matter of consensus. And that is the key point about performative paper — the uniqueness characteristic is a matter of long developed social convention. Kent, (1978), discusses similar difficulties in the context of database design.

In electronic form, the original recording of a document is indistinguishable from any of its duplicates. Indeed, what appears as the electronic movement of a document from one place to another is actually copying its information pattern from one magnetic device to another, then erasing the original. Thus, the concepts of individuality and uniqueness of an original and its copies become blurred when a document is converted to magnetic form. Our social conventions delineating uniqueness are not yet refined for electronic media.

Strawson (1959) presents philosophical discussion of the individuation problem. He observes that the entities for which we have a clear concept of individuality and uniqueness are those that can be situated, either directly or by a unique chain of associations, in the general framework of space and time.

Thus, hard, physical objects that undergo only minor transformations have a unique location in the spatial temporal framework at any point in time. More diffuse objects are more difficult to individuate. An example might be a disease. Asserting that two patients have the same disease typically means that the bacteria or virus are biologically of a common category, or it may mean that they are of a common population. The latter assertion includes a conjecture of contagion. A population has a spatial/temporal location whereas a generic type does not.

Moving into the domain of conceptual objects individuation becomes more difficult. Consider for instance a musical composition. We may know it through various performances or its various representations as printed musical scores. But to claim that any two of these are the same typically reduces down to identifying a chain of reproductions back to an original event when the piece was composed, i.e., locating it in space/time.

Other conceptual entities whose historical origins have been forgotten are notoriously difficult to individuate. For example, people typically distinguish various forms of socialism by relating them to their original authors, e.g., Marxism, Maoism. However, the various forms of capitalism are not so clearly distinguished, since the historical origins are not so well known.

In database management it is common to distinguish between *type* and *instance*. A typical example is the generic concept EMPLOYEE vs individual instances of employees, John Doe, Mary Smith, etc. The point here is that this distinction is fairly well understood in the case of physical objects, but becomes blurred as one considers less tangible entities.

The above example of music compositions is an important intermediate case. Books and other printed materials have similar individuation characteristics, namely that they are easily reproducible (Thompson 1981). Computer software and data have this feature in the extreme. Indeed, in virtual memory systems and distributed databases, a particular program or data set may be automatically copied to and from hundreds of locations without the user's awareness. It is the extremely facile reproducibility of computer media that presents a challenge to the management of performative documents, for these require non-reproducibility.

But why does originality and uniqueness of representation play such an important role in the case of performative documents? Basically, it is due to the above mentioned observation that the document serves as social evidence of someone's personal commitment to a belief, attitude, or intention. In physical form this evidence is much easier to control, e.g., I can void a check by tearing it up. The cases where this is most sensitive are when the document serves to obligate the author (or sometimes another party) to the performance of some actions, for instance, paying a sum of money. Here it is essential that the document have a unique, non-duplicable representation so that the author cannot be forced into further obligations by simple mechanical reproduction.

Note that encryption methods for producing digital signatures (e.g., Diffie and Hellman 1979) do not address this particular problem. They guarantee the identities of the sender and recipient of a communication, but do not block the reproducibility of the document once it has been received.

The major application of computer management of performative documents is the case of electronic funds transfer system (EFTS), used for financial transfers between banks and other financial institutions. Here the individuation problem is controlled by a neutral third party (the Federal Reserve in the US), which monitors the transactions and insures against illegal reproduction. This is similar to the role of a witness in verbal contracts, or to the role of a notary in other types of legal transactions. The notary function, or some analogous form of social convention, is one way of resolving the individuation problem arising from the electronic representation of performative communications. Unfortunately, this increases the amount of human overhead of the system's operation and so reduces its cost/effectiveness.

The individuation problem of performative documents is thus one involving the interaction between information technology and the sociology of organizations. Further aspects of this interaction are the subject of the next chapter.

CHAPTER 9:

BUREAUCRACIES, BUREAUCRATS AND INFORMATION TECHNOLOGY

CONTENTS

A. INTRODUCTION

Bureaucracy. The term is laden with negative connotations. One thinks of large, rigidified organizations with baroque, ritualized procedures incapable of adapting to changing needs and conditions in the environment. In mentioning the term bureaucracy one usually also speaks of its means of perpetuation: the professional bureaucrat. These are usually cast as unimaginative, plodding individuals socialized into the rule system of the bureaucracy to the point where the rules themselves, and not the purposes behind the rules, become the reason and guides of their employ. In recent years, another force has appeared that threatens to spread the phenomenon of bureaucracy even further, namely, the implementation of these bureaucratic rules and procedures in the form of computer-based administrative systems.

The purpose of this chapter is to review in somewhat more depth the nature and interaction of these three forces: the bureaucratic organization itself; the bureaucrats who populate such organizations; and the special impact of information technology on the organization's operation.

B. BUREAUCRATIC ORGANIZATIONS

The term 'bureaucracy', as both a popular and scientific term, has come to have a variety of often overlapping definitions. The definition used here is due to Weber (1956/1978). To Weber, the process of bureaucratization is a shift from organizational management based on the interests and personalities of specific individuals, to one based on explicit ***rules*** and ***procedures***. These rules and procedures are identified with ***roles*** in the organization rather than individual people. Bureaucratic organizations thus take on an impersonal, mechanical character. To Weber, this is a positive development leading to greater effectiveness and efficiency:

> Bureaucracy develops the more perfectly, the more it is "dehumanized," the more completely it succeeds in eliminating from official business love, hatred, and all purely personal, irrational, and emotional elements which escape calculation (Weber 1956/1978:975).

Bureaucracies are sometimes characterized as having a 'mechanistic,' form of administration based on fixed rules and procedures as opposed to 'organic' organizations, which rely more on individual discretion (Burns and Stalker 1961). Bureaucracies in this sense are becoming of increasing importance in both planned and free market economies though the roles are somewhat different.

In a planned economy, the rationalization of management is central to the ideology. However, to Marx, bureaucracy was a major evil to be abolished:

> Bureaucracy becomes an autonomous and oppressive force which is felt by the majority of the people as a mysterious and distant entity — as something which, although regulating their lives, is beyond their control and comprehension, a sort of divinity in the face of which one feels helpless and bewildered (quoted in Abrahamsson1977:38).

Here the term 'bureaucracy' is used in a slightly different sense from Weber, denoting government bureaucracies in particular. The relevance for Marx was that these are an important concentration of social power.

In market economies, bureaucracy seems to be regarded more as a concession to inadequacies in market mechanisms. Here we need to distinguish bureaucracy from hierarchy. Williamson (1973) discusses 'markets vs hierarchies' as a problem of economic organization. In certain cases resources are allocated via market mechanisms; in other cases they are allocated within an organizational hierarcy, which may be under either public or private control. Hierarchies become bureaucracies (in the sense used here) when their administration becomes rationalized, embodied in explicit rules. In the case of hierarchical organizations in the private sector, this rationalization process tends to evolve gradually, as the organization discovers regularity in its environment.

Governmental hierarchies, by contrast, are typically created by legislation and so become bureaucracies from the outset. Downs (1967:32,34) cites a number of factors for the creation of governmental hierarcies. One is the case of consumer goods with large 'external' costs or benefits. An external cost or benefit is one not reflected in the good's free market price—for instance, the smog created by automobile exhaust, or non-biodegradable detergents that pollute rivers. The point is that market mechanisms do not take these external costs into account in selecting an equilibrium consumption level. To compensate for these inadequacies, a bureaucracy is often created.

Another case where a free market mechanism does not operate well is with so-called 'collective goods'. These are goods with indivisible benefits; once the good exists, everyone benefits whether or not they have paid their share. An example is national defense. In a free market, each person is motivated to avoid paying his/her part; since everyone makes this assumption, the collective good is not acquired. Again, to avoid this pathology of the market system, control of such goods is given over to a bureaucracy.

A somewhat related situation arises in certain industries such as oil production or telephone services where economies of scale or patent controls create strong monopolistic tendencies. In order to protect the consumer from unfair pricing, two options have been employed, both bureaucratic. One, is to nationalize the entire industry into a governmental agency. Examples are PEMEX, Mexico's national oil company and the various PTTs in European countries. The other alternative, effectively only slightly different, is to create a governmental regulatory agency to control the monopoly's behavior, e.g., the FTC and FCC in the U.S.

The rationalization of organizations, in itself, would seem to be inherently positive and equitable. Indeed, this is the implicit goal behind most of management science and operational research.

However, there seems to be an undesirable side effect that accounts for much of the negative connotations we attach to the term bureaucracy, namely, that highly rationalized organizations apparently become inflexible and unresponsive to changes in the environment. Weber comments:

> Once fully established, bureaucracy is among those social structure which are hardest to destroy. Bureaucracy is *the* means of transforming social action into rationally organized action... the ruled, for their part, cannot dispense with or replace the bureaucratic apparatus once it exists, for it rests upon expert training, a functional specialization of work, and an attitude set on habitual virtuosity in the mastery of single yet methodically integrated functions...
>
> Such an apparatus makes 'revolution', in the sense of forceful creation of entirely new formations of authority, more and more impossible—technically, because of its control over the modern means of communication (telegraph, etc.), and also because of its increasingly rationalized inner structure (Weber 1956/1978:987-989).

One aspect — at least in market economies — for the unresponsiveness of bureaucracies is that they typically have achieved a monopolistic or protected position where they are not forced to change by competitive pressures. Nonetheless, newly elected politicians and corporate presidents often recognize and attempt to relieve the problem, though typically with little success.

Jay Galbraith (1973, 1977) offers a useful framework for analyzing the problem. A currently popular theory of organizations is the information processing view, due principally to Simon (e.g., Simon 1955, March and Simon 1958). The key concern is how the organization copes with the *complexity* of its environment, given the bounded rationality (cognitive limitations) of its managers. Galbraith extends the information processing view of organizations, to a 'contingency theory' approach. He regards the complexity of the organizations task as only one dimension of its information processing difficulties.

Another dimension is added to the organizational design problem, what Galbraith calls *uncertainty*. This refers to the degree of unpredictability of the tasks performed in the organization:

> Uncertainty is defined as the difference between the amount of information required to perform the task and the amount of information already possessed by the organization (1973:5).

The importance of this relates to the organization's ability to plan or pre-program its activities:

> The greater the task uncertainty, the greater the amount of information that must be processed among decision makers during task execution in order to achieve a given level of performance (1973:4).

Galbraith classifies the nature of the organization's overall cognitive task (as well as any of its subtasks) on a two dimensional framework of complexity and uncertainty. This may be viewed as a matrix (Figure [9.1]) characterizing the different types of cognitive tasks that organizations face. In situations of high complexity but low uncertainty, the organization is able to plan and routinize its activities. These are the conditions under which bureaucracy is most effective. In situations of low complexity and high uncertainty, by contrast, the organization is constantly being surprised by changes in the environment. Here, the most effective form of administration seems to be one that relies heavily on the discretion of its employees. Burns and Stalker (1961) use the terms 'mechanical' and 'organic' to describe these contrasting forms of administration.

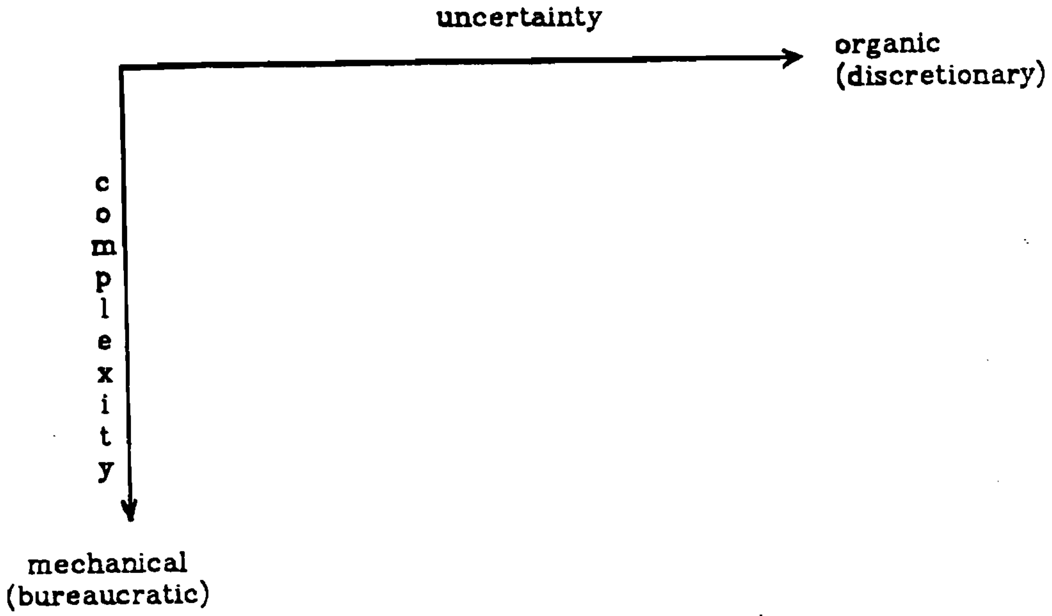


Figure [9.1].

The problem, of course, is deciding what form of administration is appropriate when the environmental demands are both highly complex and highly uncertain.

As observed, rationalization is the typical response to complexity. An apparent difficulty with rationalization, however, is that when a once stable environment becomes more uncertain, the organization seems to have difficulties de-rationalizing, that is, removing rules and procedures and relying more on individual discretion in order to become more adaptive. One factor is likely to be that it has reached a level of internal complexity that cannot be maintained in a less rationalized type of organization.

The desired response would be to move quickly to another highly rationalized configuration. However the complex of bureaucratic procedures represents a large scale intellectual effort of many people over time. Bureaucracies are not built in a day. The time required to construct a new configuration may be too long compared to the rate of environmental change.

Implicit here is the observation that the rationalization of administration and organizational adaptability *seem* to be conflicting principles. The next sections examine possible reasons why.

C. THE BUREAUCRATIC PERSONALITY

Seldom are bureaucracies discussed without considering the role played by the people who staff them. Weber for instance remarks:

> the professional bureaucrat is chained to his activity in his entire economic and ideological existence. In the great majority of cases he is only a small cog in a ceaselessly moving mechanism which prescribes to him an essentially fixed routine of march (Weber 1956/1978:988).

A bureaucrat, unlike many other vocations, is heavily socialized and hence psychologically dependent on his/her active role in the organization. Bureaucracies such as have been described generally only arise in large organizations and then usually only after a fairly long period of adjustment and stabilization. Thus the activities of a bureaucrat are not only explicitly prescribed, but their full extent and interplay with other parts of the organization is also complex and difficult to learn. The bureaucrat therefore becomes an expert in his/her role in the *particular* organization. This is for instance quite different from professionals or trade workers whose specialities are generally transferable to other organizations.

A bureaucrat's training is thus peculiar to his/her organization. This makes it unsurprising that these people cling tenaciously to their positions, building defenses and guarding informational resources to make their positions more secure.

This is one of the primary reasons why bureaucracies are so persistent. Indeed, they survive even national revolutions. For instance, speaking about the post-revolutionary period in Russia, Lenin complained:

> [During the revolutionary upheavals, the bureaucrats from the Tsaristic time had been shaken up and placed in new posts. But they did not remain there. They tried to regain their old positions.] The Tsarist bureaucrats began to enter the Soviet institutions and practice their bureaucratic methods, they began to assume the coloring of communists and, for greater success in their careers, to procure membership cards of the Russian Communist Party. And so, having been thrown out of the door, they fly in through the window! (Lenin, *Selected Works*, Vol VIII:353, quoted in Abrahamsson 1977:41-42).

These remarks relate to the complexity and specialization of the bureaucrat's training. But the socialization process of the bureaucrat is not merely cognitive, it is also epistemic. The bureaucrat does not merely understand and obey the organizations rules and procedures, (s)he also comes to *believe* in them with an almost patriotic or religious faith. This leads to a concept of 'organizational myth.' Michael (1977) notes that as regards the social/economic world, there are no scientific truths. Yet we need some coherent set of beliefs in order to plan and act. We need to have 'both feet planted firmly in mid-air.' An important aspect of a successful organization is to provide a certain philosophy or set of 'myths' that provide social unity and focus. Deal and Kennedy (1982) propose a similar concept in what they call 'corporate culture'. Here the organizational myths are enthusiastic; the image is one of growth, innovation, aggressive and spirited competition.

Bureaucracies, by contrast, have typically reached a stage where further growth and innovation are limited. The emphasis is rather on stability, correctness, and control. Bureaucratic commandments are intoned, "Thou shalt not ...." Aspiration and inspiration are tempered by the guilt of transgression. This results in what Thompson (1961:365) calls the 'bureaupathic reaction' where, strict control from above encourages employees to 'go by the book,' to avoid innovations and chances of errors which put black marks on the record. It encourages the accumulation of records to prove compliance... It encourages decision by precedent, and unwillingness to exercise initiative or take a chance. It encourages employees to wait for orders and do only what they are told.

D. INFORMATION TECHNOLOGY IN BUREAUCRACIES

Bureaucrats are no longer the only active force in bureaucracies. Whereas a bureaucrat is trained and socialized to follow prescribed procedures, a computer can likewise be programmed to follow many of these same procedures.

Indeed, the computerization of a bureaucratic process is the ultimate form of organizational rationalization. The computer is the archetype of Weber's dictum to eliminate "love, hatred and all purely personal, irrational and emotional elements" from the organization's procedures.

Yet while computers presumably help remove the undesirable caprice of bureaucrats themselves, they nonetheless have become symbols of pathological bureaucratic rigidity. We are all acquainted with the agonies of trying to rectify a computer based billing error, etc.

But is this really because the computerization of such processes actually makes them less adaptive, or is it rather that computers provide a convenient scapegoat for organizational incompetence? Systems analysts will often argue that the latter is the case. While this may be partially true, it is also true that computerization, at least in its most prevalent forms, does add to inflexibility. This stems from two interrelated problems.

The first is one of organizational responsibility: The people that use the computer programs are very seldom the ones that write them. Thus the people that are close to the problem and able to recognize needed modifications as they arise, must request the assistance of a programmer, who typically resides in a different (data processing) department.

This problem has been widely recognized and is often cited as a motivation for localized (microprocessor) computing and associated high level languages that the functional departments themselves can control; see e.g. Fick (1980). However, this is likely to be only a partial solution, applicable only to those procedures that are modular and separable to individual departments. The problem still would remain as to the management of procedures that pervade large segments of the organization, especially where these are complex and interdependent.

E. BUREAUCRACIES VS MACHINES

A characteristic of machine intelligence is that it is 'rule based'. If we consider only this software aspect (and ignore differences in processor hardware), then the most ubiquitous and successful examples of mechanical cognition are bureaucracies. Yet while the projects to create various types of artificial intelligence have a certain romance and intellectual adventure about them, the term 'bureaucracy' seems at best dreary and more often spiteful.

Consider how this view compares with standard models of computation. Recall from chapter [1], that in automata theory (e.g., Hopcroft and Ullman 1969) a computer may be regarded abstractly as a language processor, transforming an input string of symbols to output symbols. In information systems applications these symbols comprise formal language, which was called L_{RW}, containing assertions about the 'real world' (organizational environment). These assertions are normally stored in the organization's database and the processor is invoked by queries, calls to application programs, etc. Hence, what is called 'automaton' here is meant to include the entire set of application programs, DBMS software, query interfaces, etc.

The automaton, as language processor, is regarded as a grammar. This grammar is itself defined in a notation, which was called L_C. Practically, L_C corresponds to an arbitrary programming language. Ignoring efficiency considerations, L_C may be regarded as reducing to a set of production rules* of the form:

* Production systems are an abstracted notation for specifying formal language grammars. These are discussed in more detail in the next chapter.

IF <condition> THEN DO <action>.

If none of the various conditions are met, that is, if no rule is actuated, the default is inaction. The machine doesn't do anything it's not instructed to do by one of its rules.

A currently popular view of organizational management (e.g., March and Simon 1958) regards managers as information processors. Taking the metaphor literally, the automaton might be replaced with a person. The 'programming' of this person might be in another language, L_B, expressing the various bureaucratic rules and procedures this person is to follow.

But, in attempting to represent L_B (bureaucratic programming) abstractly as was done for L_C (computer programming), a problem is encountered using only production rules. As observed repeatedly in the literature on organizational psychology and sociology (e.g., Maslow 1943, McGregor 1960, Cyert and March 1963, March and Olsen 1979), people are not naturally idle. They have their own individual interests, goals, aspirations, etc., which they seek to satisfy through their participation in the organization.

When these correspond to the interests and goals of the organization itself, we tend to regard their independent behavior as 'initiative', otherwise it is considered more as the dysfunctional pursuit of 'personal interest'. L_B (bureaucratic programming) therefore contains another basic aspect. It not only orders the execution of desired behavior, but *restrains* the performance of undesired behavior. In the last chapter it was suggested that the underlying logic of bureaucratic procedures would require the operators of deontic logic, namely, (for q an arbitrary action):

O q	q is obligatory
P q	q is permitted
F q	q is forbidden.

To be adequate as a language for bureaucratic procedures, these operators need to include an aspect of contingency (corresponding to the conditions in production rules). Unfortunately, contingency is not straightforward in deontic logic, and a number of proposals appear (Hilpinen 1971/1981, 1981). Note that discretionary actions are those not forbidden, hence permitted. A 'perfect' bureaucracy, in the sense of being completely rationalized and determined, would eliminate permissions entirely. Everything would be either (contingently) obligatory or forbidden.

This is of course a macabre and unworkable design for any human organization. As Norbert Wiener (1967) once argued, such extreme regimentation is an inhuman use of human beings; such activities are not only economically but morally better left to machines.

F. CORPORATE CULTURES

The information processing views invite the comparison between (human) organizations and (mechanical) computers. However, people have a characteristic that computers (as we know them) do not have, namely *preferences* (intrinsic goals, values, drives, motivations, etc.). People may *prefer* chocolate to vanilla, computers don't.

Computer programs are composed of commands. Their behavior is described as a sequence of imperatives, where the default is inaction. However, the default behavior of people depends on their individual interests and desires. This leads to the observation that a major effect of bureaucratic red tape is not just to invoke action but also to constrain it. It is for this reason that deontic logic, rather than imperative logic, has been suggested as the appropriate model of bureaucratic authority. Subordinates are not automatons. Bureaucratic rules and procedures *restrain* rather than simply dictate their behavior. (Consider the union strategy 'work to rule', which can be nearly as effective as strikes in worker protests.)

An important aspect of bureaucracies is the substitutability of personnel. This is accomplished through detailed job descriptions, which prescribe and limit the activities of the people in these roles. It is through this device that the bureaucracy maintains a uniformity of response throughout its geographical and temporal extension. Idiosyncratic behavior of individuals is restricted in a complex of prohibitions, obligations and permissions.

In the bureaucratic philosophy, idiosyncratic behavior is regarded as bad, something to be eliminated. The implicit assumption is that this behavior will not be directed towards the organization's goals, but to purely personal ones. However, idiosyncratic behavior that furthers the organization's interest is *initiative*. This is the source of adaptation and innovation.

In the Galbraith matrix (Figure [9.1]), the unexplained quadrant in the lower right included organizations facing environments that are both highly complex and highly uncertain. Yet such organizations exist and flourish — e.g., IBM, Dupont, General Electric, as well as 'Japan Inc.'. Deal and Kennedy (1982) introduce an additional explanatory component in their concept of 'corporate cultures'. From a number of case studies of large corporations in various industries and circumstances, they observe 'strong culture' to be an important success factor.

Culture is of course a difficult variable to define. They intend it in the anthropological sense indicating a commonality of interests, beliefs, and values. Further, this is not an accidental coincidence: people identify themselves as members of the culture and accept the collective views and interests as major influences on their own. Thus in such multiculture countries as Switzerland, Canada, or Belgium, there are few remaining racial differences between the cultural sub-groups. Rather people become members of the culture at birth and are socialized to accept the local norms and habits. Amongst these dialect is an especially important aspect of cultural identification (e.g., Swiss-German vs Austrian-German vs Bavarian-German).

Identification and socialization are major aspects of corporate cultures as well. Initiation into the culture begins with employment interviews, which are often conducted with great care. Deal and Kennedy cite an example from Tandem, Inc. (to them, a strong culture company) where an employee was interviewed four times for a position as purchasing clerk. The point is that these companies screen very carefully for cultural compatibility.

Once accepted, the socialization in these companies is very strong. Aside from normal task related concerns, these companies sustain elaborate structures of corporate ceremonies, mottos, heros, and legends. The employee, in addition to membership in the social culture, is reinforced in his/her membership in the organizational culture. Thus while Americans, French, Germans, etc., each share certain similarities in mentality, work ethics and values, so too do the IBM, the Procter and Gamble, the General Electric cultures, even though they span several social cultures.

Through membership in the organizational culture, employees do not necessarily come to think alike, but rather they think *together*. Rather than simply following bureaucratically defined communications channels, the informal communication becomes an important integrating aspect. Informal socializing is a major aspect in all organizations. The key point here is that in a strong corporate culture it becomes organizationally directed.

Through socialization, the organization's goals are a strong influence on the employee's goals. Personal interest tends to correspond more closely with the organization's interest. On the other hand, the organization's interests are more likely to be influenced by the consensual interests of its employees as well. Since the employees maintain a dual cultural membership, in the organization and in the surrounding society, the employees' influence helps to ensure a more appropriate relationship between the organization and its social environment.

G. MANAGING BUREAUCRATIC SOFTWARE

The concept of corporate culture is an enthusiastic one. It has something of the flavor of a large scale football rally, complete with mottos such as 'progress is our most important product' (General Electric), 'better things for better living through chemistry' (Du Pont), and so on. However, a football team does not succeed only on team spirit. Rationalization is also important, e.g., football plays, specialized skills of the players. Likewise, rationalization is a vital complement to organizational culture. The point is that, to be effective, it mustn't supercede the culture (this applies on a societal level as well). Rationalization is a tool, a component of administration, but not the whole thing.

This suggests that rationalization is a thing to be managed, just as the organization manages other assets and technology. The information processing metaphor invites a concept of 'bureaucratic software'.* Bureaucratic software is the collection of rules, procedures, job descriptions, etc., in the organization. The issue is whether this can be managed, perhaps drawing on the experiences from managing computer software. Indeed, the metaphor converges at the level of automation in the organi-

* Dobrov (1979) has a related concept he calls 'orgware'.

zation, computerization being an extreme form of rationalization.

The advantages of a concept of bureaucratic software would be to apply such concepts as program libraries and various programmer aids to the design and maintenance of organizational rules and procedures. The eventual goal would be towards improved bureaucratic software engineering.

This raises the issue of language. Bureaucratic software at present is largely in a natural language form. However, it typically occurs in a restricted style and content, somewhat like the 'legalese' of commercial contracts or legislation. There is little poetry in job descriptions and procedure manuals. The conjecture is that a substantial part of this could be codified in a more formal language, capable of mechanical inference. It is here that mechanical aids could be developed to aid in the adaptation of bureaucratic structures.

The potential would be to increase the ability to easily modify bureaucratic rule systems in response to changing circumstances. But modifiability is not only a difficult issue for bureaucratic software but for computer software as well. This is the subject of the next chapter.

CHAPTER 10:

APPLICATIONS SOFTWARE AND ORGANIZATIONAL CHANGE

CONTENTS

A. THE PROBLEM: SOFTWARE FOR ORGANIZATIONAL CHANGE

It is a commonplace observation that organizations, to survive, must adapt to changes in their environment. Those that do not are forced out of business, if they are companies in a competitive market; have their budgets canceled,in the case of government bureaucracies; or are overthrown, in the case of governments themselves.

Just how an organization should be designed to accommodate change is, of course, a much more difficult matter, and has been the subject of many volumes of organizational theory. One aspect of this general problem seems to have been neglected, namely, the effect of information technology on the organization's ability to adapt and change.

Certainly, there are numerous clear cases where the installation of an information system adds to the organization's flexibility. For instance, the installation of a centralized database may allow data to be accessed and combined in a variety of ways that would have been practically impossible when that data was recorded in paper files scattered throughout the company.

The flexibility of a given computer application obviously depends on the foresight of its designers. To this end, programming students are generally taught to seek the most general definition of the problems they are given so that the resulting program can handle not only the immediate problem but also variants of it that might arise.

This strategy has obvious limitations. In seeking to find a generalized solution, the programmer may waste undue amounts of time on conditions that will never arise. He/she must therefore make a choice as to how much flexibility to encode into the program logic. We refer to the level of flexibility chosen as the 'designed flexibility' of the system.

Selecting the appropriate level of designed flexibility is however difficult and, almost certainly, new requirements will later arise that were not planned for originally, so that the program must be modified. This is where the problem arises.

Anyone who has written even small programs will know that it is much easier to incorporate a given feature in the program logic in its original writing rather than try to add this feature afterwards. This difficulty rises exponentially with the complexity of the original program or system. (By 'system' is meant a collection of programs and data files with interdependent functions.) Indeed, the cost and effort of modifying such systems often exceeds that of their original development. For instance, Wulf (1977) refers to:

> the extreme difficulty encountered in attempting to modify an existing program. Even though we frequently believe that we know what we will want a piece of software to do and will be able to specify if precisely, it seems to be invariably true that after we have it we know better and would like to change it. Examination of the history of almost every major software system shows that so long as it is used it is being modified! Evolution stops only when the system is dead. The cost of such evolution is almost never measured, but, in at least one case, it exceeded the original development cost by a factor of 100.

Altering existing computer systems is not only expensive, it is also risky. De Millo, et al. (1979) noted:

> Every programmer knows that altering a line or sometimes even a bit can utterly destroy a program or mutilate it in ways we do not understand and cannot predict...

Indeed, beyond expense and risk, there seems to be an eventual limit to the number of modifications these systems can undergo. Winograd (1979) remarks

> Using current programming techniques, systems often reach a point at which the accretion of changes makes their structure so baroque and opaque that further changes are impossible, and the performance of the system is irreversibly degraded. (p.392)

To summarize, the basic problem with current application systems is that they are 'brittle'; i.e. they cannot easily be reformed to adapt to changing circumstances. This brittleness has profoundly disturbing consequences as more and more organizations, ranging from small and medium size companies to immense governmental agencies, convert their information processing to computer software. The immediate gains of increased efficiency, speed of processing, rapid access to centralized data files, etc., are clear (or the investment would not be justified).

However, there may be a long term, possibly devastating hidden cost as the organization finds its ability to adapt and respond to new environmental conditions hampered by its inability to modify its information systems accordingly.

B. ANOTHER PROBLEM: TRANSPORTABILITY OF KNOWLEDGE

By 'application system' (or simply 'application') we refer to a computer system composed of various programs and data files that together perform some identifiable organizational task — e.g. sales order processing, inventory control. Our attention is therefore to the software that deals directly with the organization's operations and not for instance operating systems etc., which service the internal operations of the computer.

Applications software of this sort is by and large custom made for each organization usually by an in-house data processing (DP) department. More importantly, these applications are typically written 'from scratch'. That is, they do not make use of previously developed program code pertinent to the problem domain.

The exception to this is the use of 'off the shelf' program packages and, occasionally pre-written subroutines which the new program can call at the appropriate point. For instance, numerous packages exist to do statistical analyses and quantitative algorithms and are used quite frequently in scientific applications. Likewise, off-the-shelf packages exist to do such organizational tasks as payroll processing and inventory control. This latter class of pre-written software has, however, been less successful.

The problem, once again, has to do with the 'designed flexibility' of the package. In scientific applications, the contexts in which a particular analysis or algorithm is used is relatively well specified. For instance, in any application of a linear programming algorithm one must specify the objective function, constraints and technological co-efficients and one receives the values of the decision variables as a result. For most organizational applications, however, the problems are less standardized. Probably the most regular of these is payroll processing, but even there considerable variations may exist from one firm to another as to benefits to be added, automatic deductions, classifications of labor, etc.

In order to make use of an off-the-shelf package for such applications, the particular characteristics of the organization's problem must fall within the designed flexibility of the package. When this does not occur the DP department may sometime try to modify the package. However, the general experience is that it is usually easier and more reliable to re-program the whole thing from scratch.

We call this aspect of application software development the problem of 'transportability of knowledge' from one application to another. As observed, this is generally an all or nothing proposition. One may transport chunks of knowledge from one system or program to another only in the case that the chunk corresponds to a whole program or subroutine. There seems to be no middle ground; that is, where one could make use of an arbitrary part of one program function in developing another.

The consequence of this is that software for organizational information processing is not a smooth evolution; it does not build naturally from previous experience. Thus, for example, after a quarter century of automated payroll processing, firms still often have to write new payroll programs.

By contrast, knowledge in the form of human expertise is easily transportable. For instance, when company X hires a new bookkeeper, it is doubtful X's accounting system exactly fits the bookkeeper's training or previous experiences. However, provided the new person is reasonably competent, he/she can adapt to the new system after a brief orientation period. With applications software it is as though a complete re-education, starting with grammar school, were necessary.

We summarize the arguments thus far. The basic claim is that a fundamental problem exists in the basic architecture of applications systems, namely, that they are too 'brittle' and resistant to change. This has two important consequences. One, as discussed in the last section, is that as an organization becomes increasingly reliant on its information system, it too becomes brittle and unable to adapt easily to new situations. The other consequence, the point of this section, applies not just to individual organizations, but to information system technology at large: current software architecture does not provide the proper framework for a smooth evolution of problem solving capability. We are forced to repeatedly re-invent wheels. Progress (what little can be seen) has always been in the form of someone's coming up with a bigger wheel. That this is wasteful of money and effort is the smaller part of the problem. The deeper difficulty is that when someone finds an improved method for some organizational task, these advances cannot easily be promulgated to other software for related tasks. The industry of applications software development thus cannot build on its accomplishments, and must continually re-start from the ground.

In the sections to follow, we examine the technical reasons why applications systems are so brittle. This has two closely related aspects: the first arising from the way program logic is structured; the second due to the ways data is organized in data files and data bases. An alternative architecture for applications software will be proposed that avoids these problems, albeit not without certain costs.

C. THE PROBLEM WITH PROGRAMS: PROCEDURAL LANGUAGES VS PRODUCTION SYSTEMS

Statements in a programming language are in the form of *commands* to the machine — e.g. add this, move this data from here to there, print this on the terminal.

A computer program is thus a *sequence* of such statements, e.g.

```
10  LET X = 2
20  LET Y = 3
30  LET Z = X + Y
40  PRINT Z
```

Here, the statements have been numbered for identification purposes. Importantly, the ordering of the statements in this program indicates the sequence in which the commands are to be performed by the machine.

This otherwise linear sequence of execution can be modified by what are called 'control statements'. Consider, for instance, the program:

```
10  LET X = 0
20  ADD 1 TO X
30  PRINT X
40  IF X = 100 GO TO 60
50  GO TO 20
60  STOP
```

When executed, this program prints the numbers from 1 to 100. Here, statements 40 and 50 are control statements. In statement 40, if X has reached 100, program control jumps to statement 60 where it stops. Otherwise, statement 50 directs the program control back to statement 20 where X is again incremented, printed, etc.

Thus, the execution sequence in such computer programs normally follows the top to bottom ordering of the statements, except when superceded by the effects of control statements.

Computer languages of this type are called *procedural*. These are basically the only type used in commercial practice, and include all the well known languages for data processing and scientific applications — e.g. COBOL, FORTRAN, PL/I, BASIC, ALGOL.

In these cases, the knowledge embodied in the computer program is expressed as the specific steps for doing it. A key thing to recognize is that this procedurality makes the statements of the program interdependent. Generally (though not always) changing the order of any two statements makes a serious change to the program's operation.

While it may not be patently obvious from the two tiny examples above, it is this inter-dependence that makes computer programs so difficult to modify.

As a result of an interesting blend of computer science and formal linguistics, an alternative approach has emerged over the last decade or so. This approach is based on so-called 'production systems' (PSs) which enable the knowledge of the program to be expressed in a form that is independent of its execution sequence.

The concept of production systems was first proposed by the linguist Post in 1943 to aid in the formal specification of natural language grammars. The basic idea is extremely simple. A single production is a rule of the form:

IF <pattern> THEN <action>,

or, in the more usual notation,

<pattern> → <action>.

A production *system* consists of a 'data base' and a collection of such production rules. (This is a database in a fairly restricted sense, not to be confused with those maintained by database management systems.)

The pattern in each rule is some condition to be matched by the database and the action is typically some modification to the database. In the purest form of a production system, the rules are arranged in a linear order. Starting from the beginning the patterns are compared to the database until a successful match is found. The corresponding action is then performed and the process is repeated, starting once again from the beginning comparing the patterns to the database.

Nilsson (1980:21) summarizes this as the following generalized procedure:

Procedure PRODUCTION

1. Data ← initial database
2. Until DATA satisfies the terminal condition, do:
3. begin
4. select some rule, R, in the set of rules that can be applied to DATA
5. DATA ← result of applying R to DATA

6. <u>end</u>

Consider for instance the following example for recognizing a certain type of English declarative sentence.

1	THE → DET	8	N → NP
2	ON → PREP	9	ADJ NP → NP
3	HUNGRY → ADJ	10	DET NP → NP
4	BIT → VT	11	PREP NP → PP
5	DOG → N	12	VT NP → VP
6	CAT → N	13	VP PP → VP
7	NECK → N	14	NP VP → S

15 S → halt

The production rules on the left represent a lexicon indicating the grammatical categories of various words. The rules on the right indicate the grammar proper.

In formal grammars, a distinction is normally made between ***terminal symbols***, i.e., the basic symbols in the language (English words in this case), as opposed to ***non- terminal symbols***, which indicate grammatical constructs. However, in a production system implementation of such a grammar, these are simply different elements of the database. When the database consists only of the symbol "S", the sentence is accepted as grammatical and the system halts.

For example, suppose we have the sentence:

"The hungry dog bit the cat on the neck."

the database transformations would be as follows.

THE	HUNGRY	DOG	BIT	THE	CAT	ON	THE	NECK	initial
DET	HUNGRY	DOG	BIT	DET	CAT	ON	DET	NECK	rule 1
DET	HUNGRY	DOG	BIT	DET	CAT	PREP	DET	NECK	rule 2
DET	ADJ	DOG	BIT	DET	CAT	PREP	DET	NECK	rule 3
DET	ADJ	DOG	VT	DET	CAT	PREP	DET	NECK	rule 4
DET	ADJ	N	VT	DET	CAT	PREP	DET	NECK	rule 5
DET	ADJ	N	VT	DET	N	PREP	DET	NECK	rule 6
DET	ADJ	N	VT	DET	N	PREP	DET	N	rule 7
DET	ADJ	NP	VT	DET	NP	PREP	DET	NP	rule 8
DET	NP		VT	DET	NP	PREP	DET	NP	rule 9
NP			VT	NP		PREP	NP		rule 10
NP			VT	NP		PP			rule 11
NP			VP			PP			rule 12
NP			VP						rule 13
S									rule 14
halt									rule 15

Note that the production system would have reached the same conclusion had the ordering of the rules been reversed. This could hardly be done in an ordinary computer program. On the other hand, with the rules reversed, the system would have been much less efficient since, for instance, the initial translation of terminal symbols would have needlessly searched through the higher level transformation rules.

The initial applications of production systems in computer science were in the area of compiler theory, i.e., in specifying the syntax and interpretation of programming languages (as opposed to natural languages). Subsequently, it has been recognized that PSs have a potential much broader range of usefulness. For instance, one classic application was the Logical Theorist of Newell, Shaw and Simon (1963). Beginning with the initial axioms and rules of inference of Russell and Whitehead's *Principa Mathematica*, the Logical Theorist successfully proved all the theorems of this massive text. Indeed, in several cases it found original proofs, simpler than the original.

Another famous example of the use of production systems was Shortliffe's MYCIN system (1976). The purpose of MYCIN is to perform medical diagnosis. In this case, the database is the patient's symptoms, as revealed by various laboratory tests, etc. The production rules (some 300 of them) are thus the sort of medical deductions a doctor might make based on these symptoms. For example:

IF	the infection type is primary-bacteremia, the suspected entry point is the gastrointestinal tract, and the site of the culture is one of the sterile sites,
THEN	there is evidence that the organism is bacteroides.

Within the area of Artificial Intelligence (AI) numerous other applications of production systems have been explored.

Davis and King (1975), is an excellent survey article on production systems. Commenting on the types of applications where PSs are best suited, they observe that where the emphasis of a task is on recognition of large numbers of distinct states, PSs provide an advantage. In a procedurally-oriented approach, it is both difficult to organize and troublesome to update the repeated checking of large numbers of state variables and the corresponding transfers of control....

[PSs are] characterized by the principle that "any rule can fire at any time," which emphasizes the fact that at any point in the computation, any rule could possibly be the next to be selected, depending only on the state of the database at the end of the current cycle. Compare this to the normal situation in a procedurally oriented language, where such a principal is manifestly untrue: it is simply not the case that, depending on the contents of the database, any procedure in the entire program could potentially be the next to be invoked.

PSs therefore appear to be useful where it is important to detect and deal with a large number of independent states, in a system which requires a broad scope of attention and the capability of reacting to small changes.

With regard to the ease of modification of PSs, they continue (p.20):

We can regard the ***modularity*** of a program as the degree of separation of its functional units into isolatable pieces. A program is ***highly modular*** if any functional unit can be changed (added, deleted, or replaced) with no unanticipated change to other functional units. Thus program modularity is inversely related to the strength of coupling between its functional units.

The modularity of programs written as pure production systems arises from the important fact that the next rule to be invoked is determined solely by the contents of the database, and no rule is ever called directly. Thus the addition (or deletion) of a rule does not require the modification of any other rule to provide for (delete) a call to it. We might demonstrate this by repeatedly removing rules from a PS: many systems will continue to display some sort of "reasonable" behavior, up to a point. By contrast, adding a procedure to an ALGOL-like program requires modification of other parts of the code to insure that it is invoked, while removing an arbitrary procedure from such a program will generally cripple it...

Thus where the ALGOL programmer carefully chooses the order of procedure calls to create a selected sequence of environments, in a production system it is the environment which chooses the next rule for execution. And since a rule can only be chosen if its criteria of relevance have been met, the choice will continue to be a plausible one, and system behavior remain "reasonable," even as rules are successively deleted.

As described so far, pattern matching proceeds from the beginning of the rule set each time until a match is found, in which case that corresponding action is taken and the process is repeated.

However, in the notion of a 'pure' PS, each rule supposedly has an equal chance of firing — i.e., its position in the rule set should not affect its chances of firing. This only causes difficulty when the patterns of more than one rule match the database, in which case a choice of which action to take must be made. A variety of approaches have been used to resolve such rule contention, for instance:

rule order	—	use the first matching rule.
data order	—	data elements are assigned priority: pick the rule whose match gives the highest priority.
generality order	—	use the most specific rule.
recency order	—	use the most recently executed rule.

Recall that each rule is matched against the entire database and that two simultaneously activated rules may have matches on completely separate parts of the database. Clearly, rule contention is only problematic when the firing of one rule would disable the database match of the other candidate rule(s).

Thus, in the pure form of a PS, *all* of the rules should be tested against the database on each cycle, the subset of matching rules selected, and a choice made (by some criterion) as to which of those should be allowed to fire.

However, as the database and/or number of rules gets large, the system degrades for lack of efficiency. In consideration of this, a number of production system implementations have allowed some degree of control structure to creep back in. Thus, various strategies or 'heuristics' have been employed to increase the likelihood that, for certain contexts, the applicable rules will be found quickly and that the entire rule set need not be examined without danger of ignoring an applicable rule.

Thus, a number of PS implementations exhibit a greater or lesser degree of 'partial procedurality' as production systems augmented with a control structure mechanism. The design of such control structures, so as to provide efficient search without nullifying the advantages of flexibility offered by the basic PS orientation, has become a matter of intense interest and debate within computer science (see for example Winograd 1975; Kowalski 1979a).

This is an interesting development for the context of this discussion since it provides a framework for examining various styles of rule organization and management along a *continuum* of procedurality, instead of a flat choice between the two extremes.

A sign of the potential viability of production systems has been the rapidly increasing popularity of the language PROLOG.* Originally developed in the early 1970's by Colmerauer at Marseille, France, it has since been re-implemented and extended numerous times at universities and research institutes in France, England, Canada, Portugal, Hungary

* The name PROLOG, standing for PROgramming in LOGic, is now more of a historical acronym due to the language's construction around the "resolution principle," a technique used in automatic theorem proving. While theorem proving remains as one of the application areas of PROLOG, its usage has since broadened considerably to include relational databases, natural language parsing, expert systems, etc.

and elsewhere.

PROLOG is a 'backward inferencing' production system; i.e. PROLOG programs are written to deduce backwards from a specified goal to the available facts in the program's database. Partial procedurality may be introduced through a special device called a 'cut'. Thus, PROLOG programs may be written as purely declarative rules, without using the cut; but may be made increasingly procedural through extended uses of the cut operator.

Excellent texts on PROLOG are by Kowalski (1979b) and also Clocksin and Mellish (1981); a wide range of PROLOG applications and example programs are discussed in Coelho et al. (1980). A perceptive critique of the language for the American artificial intelligence community (which seems to be more committed to the language LISP), is given by McDermott (1980).

D. THE PROBLEM WITH DATA: DATA FILES VS PREDICATE CALCULUS

Most application software used in organization centers around the processing of large amounts of data (as opposed to, for instance, optimization routines that are much more computation intensive on relatively small amounts of data). Hence, inflexibilities introduced by the way data is organized in data files and databases are equally as important as (if not more so) those introduced in the design of procedural programs. At any rate, as will be seen shortly, the problems are highly inter-related.

A note on terminology. In the last section, the term database was used to designate the data repository of a production system. In this section, the term database will be used more in the sense associated with database management (DM). Later we return to compare the two views, at which point they will be distinguished as PS databases and DM databases.

For the moment, however, we consider a *general* view of data maintained in data processing applications, whether this data is accessed through a database management system or not. The term 'data file'* will therefore be used to indicate a conventional data processing file or a logical segment of a database (e.g. the tuples of a single relation in a relational database; the instances of a single record type in a CODASYL database). The term 'database' will then be used to refer to a collection of such data files with inter-related subject matter (e.g. sales file, inventory file, back-order file), whether or not the access to these is coordinated by a DBMS.

Data files are usually organized as a rectangular table with labeled columns called 'fields'. For instance, a file on employees might have fields for the employee's name, address, age, salary, etc.

* The term 'data file' here corresponds to 'relation' as used in chapter 2. There, the intention was to link relational database theory to predicate logic as a basis for investigating the formal semantics of databases. Here, while the connection to logic is also made, the concerns are more practical, relating to information system modifiability. The term 'data file' is therefore used to refer to actual types of file organizations, whether or not they conform to the Relational Model.

EMPLOYEE FILE

Name	Address	Age	Salary
Adams	5 Pine Street	30	20,000
Peters	101 Broadway	45	18,000
Smith	3 Park Place	37	24,000

Sometimes data files have more complicated organizations — e.g. some columns may have multiple entries for a given data item. This tabular view is sufficient for present purposes, however. Also, this is the basic view maintained by the more popular database management models (i.e., Network, Relational).

Note that each data file has three levels of description: the ***data file name*** (e.g. EMPLOYEE), the ***field names*** (e.g. NAME, AGE), and the ***data values*** (e.g. Smith, 37). It is important to note also that a data file represents a ***model*** of some aspect of the organization, in this case, what are considered to be the important features of employees.

The structure of the data file often carries certain implicit information as well. Often, as in this example, each row of the data file implies the existence of some entity in the environment, in this case an employee associated with the company. The converse assumption is also sometimes made, e.g. if a person's name does ***not*** appear in the file, then he/she is not an employee.

Other data files, however, might have different existence assumptions. Consider, for instance, a file for parts inventories:

PART FILE

ID#	Color	WT	QTY
3	R	10	200
12	B	8	65
7	W	13	0

This file indicates the identification number (ID#), color, weight (WT) and quantity (QTY) on hand of various manufactured parts. In this case, each row of the file does not imply the existence of a part, but only elaborates the features of each generic part type. The existence of actual parts is instead indicated by the QTY field.

These might be called the existential assumptions associated with a file. Other assumptions refer to the possible data values that may appear in a given field, e.g. that SALARY must be less than 50,000.

The basic point, however, is that the data file structure itself is not sufficient to convey all these assumptions. Instead, these appear in the logic of the programs that interpret these data files. Thus, the model of the organization represented in the application system is found not only in the data files but also in the code of the various application programs. This is a problem that has been recognized for some time in database management, and has led to a number of proposals for the separate specification of so called 'data base constraints', conditions that the data in the database must always fulfill. Such constraints are maintained in a separate table, and verified by each updating program. However, these approaches do not go far enough. There is a basic problem that remains, which has to do with the very notion of 'data' itself.

In *all* data processing files and database management systems, there is a distinction between *data structure* and the data itself. What we have called the datafile names and field names, are the data structure elements of the view presented here. (Other views of data may have further structural elements.) Thus, for instance, in the above data file for parts, we have in the first row: COLOR = "RED", where the three character string "RED" is the value of the field COLOR. The point is that these data values are regarded as *strings of characters rather than as properties of objects in the environment*. Viewed only as character strings, one is unable to specify even very commonplace inter-relationships between these properties: for instance, that if a thing has a color, it must be a physical object, hence, having weight, physical extension, geographical location, etc.

The basic problem is that the variables in data management models range over sets of *character strings* (so-called 'attribute domains' in the relational model), rather than over *objects* in the environment. For instance, a database constraint that all parts are either red, blue or white would look something like:

PART.COLOR = "RED" OR "BLUE" OR "WHITE"

To recognize that these are properties of objects in the environment, a predicate calculus notation might be used, introducing the variable x to range over these objects:

1. $\forall x$ PART (x) $\rightarrow$ RED (x) V BLUE (x) V WHITE (x)

(the symbol "$\forall$" is read "for all"). The point is that in this form, one can begin to elaborate more general properties, i.e., not just of parts, but of anything that has a color.

2. $\forall x$ RED (x) V ORANGE (x) V YELLOW (x) V GREEN (x) V ... V BLACK (x) $\leftrightarrow$ COLORED (x)

3. $\forall x$ COLORED (x) $\rightarrow$ PHYSICAL-OBJECT (x)

4. $\forall x$ PHYSICAL-OBJECT (x) $\rightarrow$ $\exists n$ $n > 0$ & WEIGHT (x) = n.

(the symbol "$\exists$" is read "there exists").

Statement (2) is a disjunct of all color names used in the organization, indicating that any of these implies the general feature of being colored, and vice versa, that being colored implies one of these properties. Statement (3) says that anything that is colored is also a physical object (though some physical objects — e.g. glass, mirrors — may not be colored). Statement (4) says that for any physical object there exists some positive number that is its weight (presuming some unit of weight measure).

The direction intended by this example should begin to become clear. Reconsider the problem of transportability of knowledge discussed above. Clearly there are many commonplace connections between properties that any organization would agree upon — e.g. the simple physics of colors, weights, physical extent. These rules will hold for any physical object, from peanuts to box cars. Other classes of properties might be restricted to a particular social system — e.g. the number of spouses an employee might have, whether dual nationalities are recognized. Other classes of properties pertain to specific industries within a given social system — e.g. the accounting practices for banks vs. those for educational institutions. Lastly, there are clearly those properties that are organization specific, such as the ranks of personnel or the parts it manufactures.

Ideally, the inter-relationship of properties at any one of these levels should only have to be developed once — e.g. commonplace physics by a national or world wide bureau of standards, accounting practices by an industry accounting board. Then, the task of any particular organization in developing its application software would only be to specify the *differences* of its local practice from that of the standardized models.

The proposal here is, therefore, to offer a predicate calculus (PC) notation as a replacement for the usual data structure view, with the claim that it provides a richer framework, capable of specifying the inter-dependence of properties of objects, not just structured organizations of character strings.

It should be mentioned that this is not necessarily a recommendation that facts about the environment actually be *stored* in this form — the underlying implementation might actually make use of a more conventional data management model — but rather that the top-most *level* or *view* of the database have the PC form.

E. COMBINING THE APPROACHES: PRODUCTION SYSTEMS AND PREDICATE CALCULUS

The point of the previous section was to recommend a predicate calculus notation as a richer form of data representation. Previously, a production system approach was suggested as a more flexible framework for specifying the potential *actions* of an application system. The final step in the proposal here is to combine these frameworks, i.e., to use the predicate calculus form of database as the database of the production system.

This is essentially the approach used in chapter [2], based on logic programming. Whereas there the motivation was theoretical, to provide a bridge between relational database theory and formal language semantics, here the suggestion is that logic programming can be of considerable practical interest as well.

The problems initially set forth were twofold: the difficulties involved in modifying applications software in response to organizational change; and the problem of 'transportability of knowledge', i.e., the difficulties of using parts of previously developed software in the development of new systems.

Logic programming, by its non-procedurality and extreme modularity, offers a direction for relieving these problems. The trade-off, however, is that logic programming is computationally much more expensive than conventional methods. On the other hand, the cost of microprocessors continues to fall drastically. Indeed, these factors seem to underly the Japanese plans for developing the so-called 'fifth generation computer', having logic programming oriented hardware (Moto-oka, 1981).

Whereas these developments promise improvements to the adaptability of computer software, they do not constitute a complete solution to the problems of organizational change. This is the subject of the next and final chapter.

CHAPTER 11:

TOWARDS A THEORY OF MANAGEMENT DECISION SUPPORT

CONTENTS

A. INTRODUCTION

In this chapter we attempt to summarize and integrate the various observations made throughout the book. We believe they add up to an argument that the metaphor comparing managerial cognition to mechanical symbol processing, while useful, is also limited. This limitation centers around the dynamics of organizations: management will always be a human activity as long as societies, hence organizations, continue to change. Herein, we think, lies a theory of management decision support systems. It is a humanistic theory, based on what we view as complementary advantages of human (organic) vs mechanical cognition.

The theoretical position of DSS can perhaps best be outlined in contrast to the theoretical interests of database management (DM), operations research* (OR) and artificial intelligence (AI). DM is concerned mainly with descriptive representations. The emphasis is on simple yet computationally efficient data structures for representing organizational facts. OR and AI, by contrast, are concerned mainly with inference. OR relies on quantitative inference whereas AI is more concerned with qualitative aspects.

The position of DSS is emphasized by distinguishing three senses of the term, 'model'.

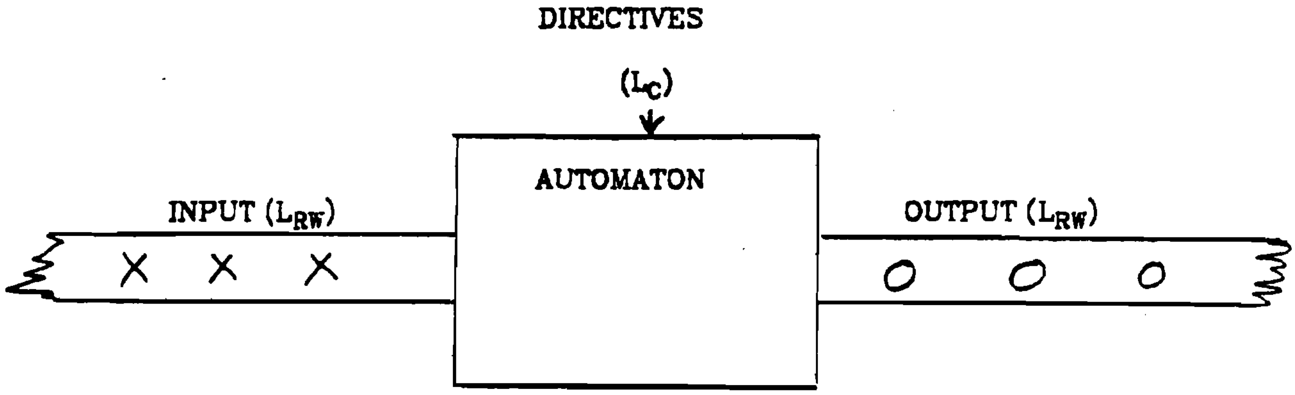
In DM the term 'data model' is used to indicate the set of representational constructs (vocabulary, formation rules) used to define a database.

* The US term 'operations research' used here, is somewhat more restricted than the British term, 'operational research'. Whereas the former focuses mainly on applied mathematics for industrial problems, the latter includes as well human engineering and organizational considerations.

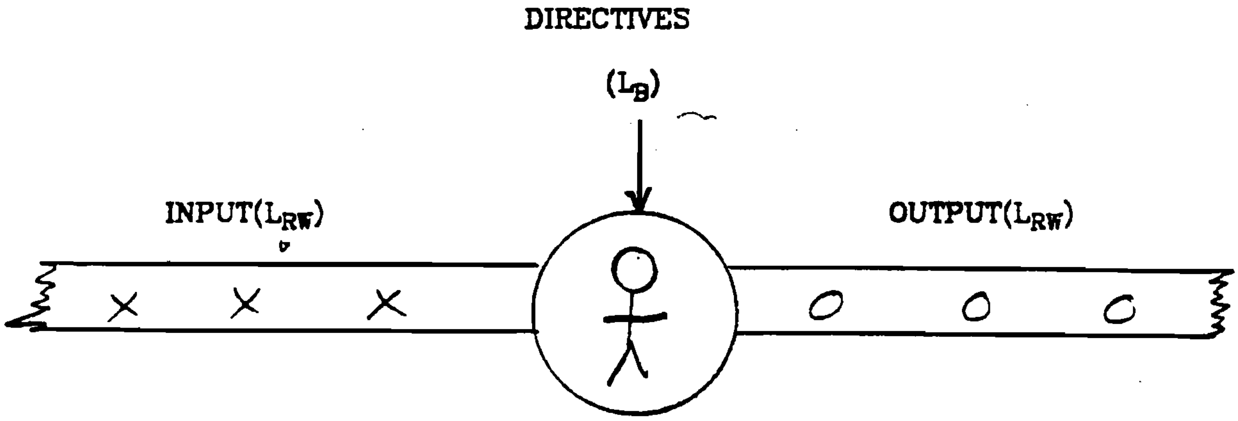
An OR model, by contrast, is a mathematical algorithm whose deductions are asserted to correspond to differences in magnitude along selected dimensions of a specified real world situation. In AI the term model is used more in the sense of a 'psychological model'. The behavior of a computer program is designed to reproduce human information processing behavior. Thus both OR and AI are concerned with inferential models, but differ in their desire to emulate human cognition.

Besides descriptive and inferential models, another sense of model has been emphasized, namely a ***semantic model*** or ***interpretation*** of a formal symbolic system. This sense complements the other senses of model in that DM, OR and AI models all depend on particular interpretations. The suggestion is that the theoretical foundations of DSS be distinguished by its concern with this third, semantic sense of model.

In the remainder of this chapter the observations in the preceding chapters are related to this formative theory of DSS. In order to help organize these arguments, the framework introduced in chapter [1], in which language and cognition is contrasted in humans and machines. For convenience, the diagram is repeated in Figure [11.1]. The diagram has two parts, comparing managers with an automaton. In part (a) a standard conception of a Turing machine is drawn having an input and output. The symbol stream of this automaton is presumed to constitute a formal language, L_{RW}, describing the real world. L_{RW} therefore corresponds to the data stored, updated and retrieved in the organization's databases. This language is distinguished from L_C, the computer language for programming the automaton.



a. Mechanical cognition



b. Managerial cognition

Figure [*11.1*].

In part (b) of Figure [11.1], the role of the automaton is substituted by a human manager, or perhaps a team of managers. Again there is a language, L_{RW}, describing the real world, which these managers process. In this case L_{RW} is a natural language, though it may contain formal language components. (Recall that formal languages have explicit and fixed rules controlling their syntax and semantics. Natural languages may evolve, depending on the consensus of the linguistic community.) Corresponding to L_C, the computer language used to program the automaton, managers learn their duties in L_B (for 'bureaucracy'), the language of their job descriptions and other directives.

The subsequent discussion is divided into two parts. The first part focuses on the language L_{RW} and issues relating to semantic change in this language. The second part is concerned with fundamental differences between human vs mechanical cognition, reflected in differences in their respective command languages, L_B and L_C.

B. SEMANTIC ASPECTS OF L_{RW}

In chapter [1], the distinction was made between L_C, the language referring to the computer and its operation, and L_{RW}, which referred to the organizational environment. In current terminology this might be phrased as programming language semantics vs database semantics. Our concern here is with the latter. As before, we attempt to avoid the present debates (e.g. various data management models vs semantic network representations) by skipping over aspects of psychological modeling, retrieval efficiency, etc. and assume that L_{RW} can be characterized as a (first-order) predicate calculus language.

The other advantage of this assumption is that it helps to focus the immense literature on formal semantics without computational distractions. In the predicate calculus (data management and semantic nets as well) we typically make the assumption that semantics follows syntax. That is, the semantics of complex expressions is constructible from the semantics of its syntactic constituents (Dowty et al. 1981:Ch. 2). This is Frege's 'Principle of Compositionality'.* The role of the usual logical connectives and quantifiers in constructing the semantics of first order assertions is well studied (van Fraassen 1971). What remains is the semantics of the open vocabulary of the logic, namely individual and predicate names. The approaches at this point divide roughly into two camps — what will be called the ***extensional*** and ***intensional*** viewpoints.

* Here we are speaking of formal, constructed languages. The principle of compositionality doesn't always hold in natural language, e.g. for proper nouns like 'Marilyn Monroe' or nominal compounds like 'red herring', where the referent of the expression is not constructable from the referents of its component words.

1. Extensional Semantics

The extensional viewpoint is dominant in formal logic, originating mainly from the model theory of Tarski (1956). Here, individual objects are regarded as primitive, leaving generic properties and relationships to be defined set theoretically. An interpretation or *model*, of a given (first order) predicate logic therefore begins with the assumption of a domain of individuals, D, and an interpretation function, F, which maps individual names to individuals in D, 1-place predicates to subsets of D,. n-place predicates to relations on D, etc. Hence a model M of a language L has the form

$$M_L = <D,F>.$$

This is entirely satisfactory as long as the population of individuals in D can be clearly specified, and they don't change.

However, a problem for management applications is that organizations and their environments *do* change. Change is fundamental to economic growth; it can't be ignored. An obvious step is to extend the model to include a time dimension, T, so that D includes all individuals existing at different times. Models of the language are then of the form:

$$M_L = <D, T, F>$$

This, however, encounters difficulties when we consider aspects of the *future*. Much of management is concerned with planning. Since there may be a variety of alternate or contingent plans, we must likewise consider multiple futures. This leads to another extension to the model including so-called possible worlds, W, hence adopting models of the form:

$$M_L = \langle D, T, W, F \rangle$$

This is essentially the ontology proposed by Montague (see Dowty et al. 1981, Lee 1981). While this enables a mathematically elegant solution, the question is whether it is still semantics. If semantics is the correspondence between symbols and the world, but if the world is not merely the actual world (past and present) but also future and hypothetical worlds, we have to consider how it is we know about these other worlds.

Strawson (1959) points out that the principle basis for our shared epistemology is reference within a common spatial/temporal framework. Possible worlds are mental constructions, Gedanken experiments. They are outside the framework of external reference and so are questionable as a basis for mutual understanding. We return to this problem shortly.

2. Intensional Semantics

The intensional viewpoint is more characteristic of the AI paradigm (especially semantic net representations). Here, it is not individual objects that are primitive, but rather generic properties and relationships. Particular objects and events are seen as instances of these generic concepts. For example, we postulate primitive concepts, MALE, FEMALE, SPOUSE, CHILD and from these are able to define the entire vocabulary of kinship relations. Particular cases of family trees, etc., are regarded as 'instantiations' of these generic concepts.

The intensional approach is most satisfactory for what might be called idealized or artificial subject domains, where the scope of variation is fixed theoretically or by explicit rules. However, the intensional approach also has difficulties, especially in describing real world domains where no theoretical foundation exists. For example, suppose we want to develop a concept, LEMON. We then seek to elaborate the essential properties of lemons. This might be a property list something like:

COLOR:	YELLOW
SHAPE:	OVAL
TEXTURE:	BUMPY
TASTE:	ACID

The problem, typically, with real world domains is that we can't simply *define* what a LEMON is, but rather our definition has to correspond to what the users of the system conceive lemons to be. Now we run into the so-called 'criterial properties' problem. We want a set of properties that in conjunction uniquely selects out lemons and only lemons from the various objects in the environment. The problem here is twofold: that too many things qualify (e.g. yellow limes) and the definition excludes atypical lemons (e.g. green lemons, lemons that aren't oval). Wittgenstein (1953/1958) is a classic elaboration of these difficulties.

3. A Sociological View of Semantics

Both the extensional and intensional approaches to semantics suffer epistemological difficulties, especially in the social/economic domains typical for management. This leads to an examination of the mechanisms by which we come to know and use the terms of our everyday language.

If we follow the extensional approach, then our main focus will be on our knowledge and identification of individuals (people and objects). This brings attention to the semantics of proper names and the identification codes we assign to machines and other objects. As Kent (1978) points out, these are of fundamental concern in data processing applications, mapping database records to inventory, equipment, personnel, customers, suppliers, etc.

How are these names associated to individuals? In the case of manufactured objects, quite often the identifying name is stamped directly on the object. In the case of names of persons and companies, the identification relies heavily on honest reporting of their names by the entities themselves, e.g. on employment applications, sales orders, etc. The point is that the organization doesn't have to *recognize* these individuals through some collection of identifying properties, it is simply *told*, e.g. "I am John Doe," "Here is the XYZ company."

The point applies much more broadly. Most of what we know about other individuals (people, places, things) that are temporally or geographically distant is what we have been told. The proper name provides a tag to which various characteristics are attached. The names themselves are passed from one person to the next in a series of 'causal chains' of reference, leading back to a direct identification of the individual. Sometimes, in the case of multiple names for the same individual, the causal chains may separate, leading to assertions like

Mark Twain = Samuel Clemens

having an informative content rather than being a tautological identity.

Kripke (1971, 1972) applies this concept of causal chains in a forward fashion in characterizing possible worlds. "Possible worlds are not far-away planets," they are rather ***constructed***, based on known, actual references.

Consider, for instance, a scenario beginning with the supposition that Ronald Reagan is bald. The question arises, how do you know it's Ronald Reagan if, in this possible world, he has different properties. (We can exaggerate the case — suppose Ronald Reagan is really a robot, manufactured on Mars, etc. — this is called the problem of 'trans-world identification of individuals'.) Kripke's point is that we don't have to ***recognize*** Ronald Reagan in this world, we ***stipulate*** that he is the same in our construction of the scenario. The proper name Ronald Reagan is a 'rigid designator'.

As discussed in chapter [4], this leads to an explanation of our understanding of generic concepts like 'lemon' and 'chair'. What is socially understood by the concept 'lemon' is a cooperation between the specialized understanding of cooks, bartenders, food sellers, fruit growers, agronomists, botanists, etc. In this case, biological science serves as the defining authority of the natural kind, 'lemon'. In the case of 'chair', the chain of reference leads back, not to scientific authorities, but rather to certain chair manufacturing companies. But how do they know what a chair is? They *specify* that their products are chairs. Thus one enterprising company may stuff burlap bags with shredded styrofoam and market it as a 'pillow chair'. Another might fold and paint pieces of cardboard selling them as 'throw-away chairs'. The success of their marketing also succeeds in modifying the concept of chair.

The effect of these arguments is to introduce a sociological conception of semantics, what Schwartz (1977) calls the 'new theory of reference'. It gives a convincing account of why semantics is so difficult to do computationally: semantics isn't fuzzy, it's social. For many of our terms, e.g. lemon, chair, the extension of the concept is quite exacting. A thing is a lemon (chair) or it is not. However, the cognition that makes this discrimination is not an individual one, but rather a cooperation of a broad social network. As Putnam observes, we tend to regard words like hand tools that we use individually. For many words, a more fitting metaphor is to compare them to a big ocean liner that requires a crew of hundreds for its operation.

4. Semantic Change

Semantics is the mapping from a symbolic representation to its referrents. The concerns of the past section reflect not only the relationship between individual terms and their denotations (e.g. LEMON and all lemons), but the relationships between these individual terms to others, e.g. the use of lemons in cooking, their medicinal aspects, the marketing and distribution channels for lemons, the growing of lemons, the genetic aspects of lemon varieties. It is this range of aspects that requires a whole social network for complete understanding.

However, the knowledge of a concept at any particular node could well be formalized and subject to mechanical inference. Indeed the successes of computers to date reflect this possibility.

The problems considered here are the mechanisms by which semantic change are introduced and conveyed throughout these cognitive networks, and the consequences of this for computer aids to these cognitive processes. To repeat an earlier observation, the role of semantics in computer applications is to validate computational inferences. If the computer has a rule

$$\mathrm{FRUIT}(x) \leftarrow \mathrm{LEMON}(x)$$

the inference (modus ponens) that any particular lemon is also a fruit is true if and only if the denotation of the term LEMON is a subset of the denotation of the term FRUIT. If, for reasons of organizational or social interest, the semantics of these terms change so that the subset relationship no longer holds, the computational inference may no longer be valid.

We, as people, make words mean what we want. Language is not an abstract entity, existing independently. It is a social artifact, a behavioral convention, that has been found to be pragmatically useful.

At the micro-level are so-called 'baptism' events. I can *name* my cat George or Ludwig or whatever. If someone asks, I tell tham the name I chose. They accept that name as a matter of pragmatic convenience: it's hardly worth our time arguing over it. Further, since I own my pets, I have a social right to do the naming. Thus for instance when I get a telephone installed, the phone company *tells* me the name (telephone number) for it. They, in this case, have the baptism right. (A computer might actually select the phone number in question based on an algorithm, but the system of phone number assignments is nonetheless the social right of the phone company.)

Now consider generic terminology. When a scientist makes a new discovery, invents a new process, etc., he/she or the research group *assign* a name to it. It is promulgated via the patent registration, research publications, academic conferences, etc.

Likewise, in the economic domain, when a company markets a new product, they invent a name for it. If the product succeeds and is widely sold, it becomes a 'household word', and is incorporated into the social vocabulary. Sometimes, if the product dominates competitive products, the brand name is used generically, e.g. Coca-cola, Xerox machines, Kleenex, IBM cards.

Some people, then, invent new vocabulary. Other people choose whether or not to accept it. Not all novel research is accepted for publication; many marketing innovations fall flat. It is a matter of pragmatic consensus.

The point is underscored by a distinction by Wittgenstein (1953/1958) between *observing* and *deciding* the meanings of words. For much of our vocabulary, we are passive observers of the semantics. We accept the dominant conventions established by traditional usage. In some situations however, especially where we want to refer to some phenomenon or idea where there is no term with the designation we want, we need to change the language. We either do it by creating a new term or by modifying the designation of an existing term. Thus, to use one of Wittgenstein's examples, "What if the diviner tells us that when he holds the rod he ***feels*** that the water is five feet under the ground? or that he ***feels*** that a mixture of copper and gold is five feet under the ground?" (1958/1965:9). The term chosen in these cases is typically related in meaning. When the altered usage is temporary, limited to a specific context, we call it metaphorical. But metaphors often catch on and become part of the popular vocabulary.

Thus, in deciding the semantics of terms there are the two components of *proposing* and *accepting* the new meaning.

The essential point to be made here is that the mechanisms of semantic change rely on social pragmatism. Words change usage and new terms appear when people want to talk about new things or ideas. It depends on their evolving interests and needs. Correspondingly, the change is accepted by others depending on *their* interests and needs. New meanings are negotiated, like new products on the marketplace.

The point is that we, as humans, have the social *privilege* to buy and sell new meanings. My dog for instance growls at automobiles. However, I don't for a minute consider allowing that usage into my vocabulary. Dogs don't count as linguistic innovators. However if a great social philosopher should growl in the same way at automobiles (say, perhaps, accompanied by a convincing argument how they were the root of all evil), I might reconsider the usage and perhaps adopt it myself. Computers, as well, don't count as linguistic innovators. When a computer program bombs and starts printing garbage, I regard that as a mistake, not as new vocabulary.

The argument is that machines won't participate in semantic change, insofar as they *decide* it, for political reasons. (Political in the broad sense, e.g. the negotiation of interests and values.)

But the question remains whether machines are able to *observe* semantic change, and thus *learn* to modify their inferences accordingly. Certainly we humans learn the meanings of much of our vocabulary not ostensively (someone pointing to an object and saying its name), but rather through inferring its meaning after hearing it used or reading it in various contexts. We often detect semantic change in a similar way. The issue is whether computers would eventually be capable of detecting and learning changes of meaning in this way. This is essentially the issue of computer learning, currently regarded as the new frontier in artificial intelligence. Learning, in turn, relies heavily on *induction*, a long standing epistemological problem. Induction is normally contrasted with deduction. The distinction is central not only to language learning, but also to the formation and verification of scientific hypotheses. It is likewise central to statistical inference, which provides numerous examples. We may think of the difference in terms of the statistical concepts of population and sample. Deduction is making inferences about a sample based on known characteristics of the population as a whole. Induction is the reverse, making inferences about the population, based on observed characteristics of a sample. For instance if we have a jar full of white beads, we deduce that any sample we draw from it will also contain only white beads. However, if we don't know the color of all the beads, and draw one or more samples from it, we may *induce* that the whole population is white. However, in the latter case we are not absolutely confident as in the former case. The concept of proof in mathematics and logic is a deductive one. Whereas deduction is 'truth preserving', leading from true premises to true conclusions, with induction we need a weaker concept like 'degree of confidence', 'warranted assertability', etc.

In our ordinary experience, we do inductions all the time, e.g. we induce that the sun will rise tomorrow, we induce that a runny nose means a cold is coming on, we induce that so-and-so has a grouchy personality based on a few conversations with him. These inductions, while extremely useful as heuristics, are sometimes wrong and have to be modified: we may have allergies rather than a cold; so-and-so may not really be a grouch, he only had a few bad days. We induce the meanings of new terms and changed meanings of familiar terms in a similar way. We form a hypothesis about them and verify that hypothesis by its consistency with other usages.

The mechanisms by which we do these inductions are the research domain of psycho-linguistics. It is for instance the central issue in explaining how children learn their first language. Many deep problems and mysteries remain. That is not to say that they couldn't eventually be understood in formal terms and made computational.

These remarks lead us to consider the cognitive characteristics of humans vs artificially intelligent machines, the subject of the next section.

C. HUMAN VS MECHANICAL COGNITION (L_B VS L_C)

1. ATOMISTIC VS HOLISTIC PERCEPTION

One of the major contributions of Artificial Intelligence, perhaps, is to point out not only those parts of human cognition that are easily emulated by symbolic processing, but also those aspects that are difficult to emulate. One such aspect is that computers seem to have difficulty in seeing the woods for the trees.

For example, visual pattern recognition has been an area of considerable research. In order to discriminate relevant details, visual information is collected in tiny discrete units that are later integrated to form a larger image.

From this perspective it is astounding how people, even babies, are able to recognize a wide number of faces despite variations in expression, lighting, etc. Moreover, this recognition often ignores what would seem to be key details, e.g. we often cannot remember whether a casual acquaintance wears glasses; we sometimes do not notice when a friend shaves off his mustache or even his beard.

There have been a number of AI proposals to reproduce this more holistic type of perception. The best known is the frame approach due to Minsky (1975). A frame is a computational construct for unifying sensory data into a conceptual whole. The sub-parts of the frame are called slots, corresponding roughly to predicates and/or measurements. For purposes of recognition, certain slots may have more weight than others. Other slots are perhaps not necessary for recognition and may have default values if no sensory data is available (as when vision of an object is partially obstructed). Thus in recognizing human faces, glasses and moustaches may have relatively low weight since these aspects may change.

The frame concept was developed for applications to computer vision, but the general idea is extendable to other areas, e.g. tactile sensation, sounds. There are also analogous representations to frames for structuring observations that are not sensory based, e.g. the 'scripts' of Shank and Abelson (1977). Keeping for the moment with sensory perception, it is important to note that frames require the translation of sense data into linguistic data. It is here that semantic issues arise, for our senses discriminate far more than we have vocabulary to describe it. Consider for instance having a portrait made of your spouse and giving the artist only a verbal description of his/her face. Would the result be recognizable? Sartre, in *La Nausee*, provides an apt example of the difficulties in completely capturing sensory experience in a verbal form:

> Black? I felt the word subside, empty itself of its meaning with an extraordinary speed. Black? The root *was not* black, it was not the black there was on that piece of wood -- it was ... something else: black, like the circle, did not exist. I looked at the root: was it ***more than black*** or ***almost*** black?
>
> It ***resembled*** a colour but also ... a bruise or again a secretion, a yolk -- and something else, a smell for example, it melted into a smell of wet earth, of warm, moist wood, into a black smell spread like varnish over that senewy wood, into a taste of sweet, pulped fibre. I didn't *see* that black in a simple way: sight is an abstract invention, a cleaned-up, simplified idea, a human idea. That black, a weak, amorphous presence, far surpassed sight, smell, and taste. But that richness became confusion and finally ceased to be anything at all because it was too much. (1938/1965:186-187)

It is the ambiguities that arise from mapping rich sensory experience onto a limited vocabulary that forms the basis of so-called 'fuzzy' logics, e.g. Zadeh et al. (1975). While the semantic issues motivating fuzzy logic are generally recognized (Wittgenstein, 1953/1958), debate continues as to the sufficiency of fuzzy logic to capture the inferential patterns that follow from our perceptions.

2. Rule-Based Inference vs Expert Muddling

Expert systems are sometimes called 'rule-based' in that they typically rely on a knowledge base consisting of independent rules or productions of the form:

IF <pattern> THEN DO <action>.

The <pattern> is some condition to be matched in the database. The <action> may be inferential, i.e., updating the database, or referring to some external device such as printing results or moving the arm of a robot.

As discussed above, expert systems have had some promising early successes in such application domains as medicine, chemistry and geology. Joseph (1982) for example writes hopefully of the possibilities of (human) intelligence amplification through the use of artificially intelligent machines. However the question remains open as to how these developments will combine with human abilities to attain higher capacities of cognition.

Artificial Intelligence research has always emphasized the similarities between the observable evidence of human cognition and the behavior of computer programs. The comparison has been used in two ways: one, as a psychological methodology, using computer programs as a possible model of human cognition; the other as an engineering orientation, using human cognition as a model for building smarter computer systems.

However, by accepting these similarities as the basis for combining computers and humans in a single category of 'cognitive entities,' we are likewise led to focus on their *differences* as well.

On the one hand, there is a fairly well developed literature (e.g. Miller 1956, Tversky and Kahneman 1974, Simon 1981), which emphasizes the limitations of human cognition with respect to machines. These deal mainly with the limitations of short term memory, coupled with relatively slow sequential processing capability, which lead us (humans) to simplify problems by abstracting their components into larger 'chunks,' and using short-cut heuristics to trim down the problem's complexity.

On the other hand, another literature is emerging (e.g. Weizenbaum 1976, H. Dreyfus 1979, S. Dreyfus 1982) that emphasizes the limitations of computational cognition as compared to that of humans. The general criticism is that computational techniques rely on atomistic representations of data and the sequential application of separate and exact inference rules whereas human (organic) cognition appears to store holistic impressions and images and is capable of fuzzy pattern matching between them, which allows for greater flexibility of association.

This suggests a theory of cognitive complementarity between human and machines. Humans for instance require a great deal of discipline and training to perform the types of iterative calculations most easily programmed in machines. Contrariwise, the types of cognition that are basic even to human infants — e.g. recognizing faces and voices, acquisition of language — present deep, unsolved problems for computational theories.

The middle area where humans and machines appear to be on comparable footing is in so-called 'rule-based' systems, which form the general architecture of expert systems applications (see Davis and King, 1975, or Nilsson, 1980, for background).

Rather than procedural programs where the computer executes instruction after instruction in a pre-determined order, these are non-procedural programs of un-ordered rules where the machine searches repeatedly through the rule set for the appropriate rules pertaining to a given situation. The non-determinism of this approach sacrifices much of the efficiency where the computer normally has advantage over the human. On the other hand, it provides considerably more flexibility and adaptability, which are the human's normal advantage.

Stuart Dreyfus (1982) makes some interesting observations regarding rule-based cognition in the formation of human expertise. His claim is that the use of a small set of discrete rules is characteristic of the novice stage in the development of a particular skill. As the individual becomes more experienced, these rules are gradually refined to incorporate numerous exceptions. Additional experience adds a context dependent organization to the rules as well as additional refinement so that the rules take on a much broader, parametric character. In the case of more advanced expertise, the individual rules give way to more holistic patterns, which are no longer processed in sequence but rather in a simultaneous pattern oriented manner. Dreyfus suggests mundane examples such as learning to drive a car or playing chess. The novice driver learns to shift at specified velocities, has certain fixed procedures for parallel parking, etc. Experienced drivers, on the other hand, no longer rely on these elementary measures but rather incorporate a wide variety of factors such as the sound of the engine, road incline and surface condition, weather, and anticipated traffic situations. A key point is that at this level, most experienced drivers can no longer specify the individual factors and rules they use.

Likewise most novice chess players begin with a simple point valuation scheme for each of the players and evaluate the value of an exchange through this numeric comparison. Subsequent development adds consideration of the relative position of pieces and their projected positions through scenarios of play and counter-play. Evidence of master level chess play however suggests a much more holistic orientation depending on comparative 'field of force' in actual and potential configurations of the pieces.

The general hypothesis here is that the major impact of rule-based, expert systems will be at these types of cognition characteristics of the early to middle level stages of human expertise development.

These arguments are especially apt in the case of expert managers. Evidence for this is found in the role that formal training, e.g. from a Masters of Business Administration (MBA) program, plays in actual management practice. Having an MBA is certainly advantageous for advancement to higher management levels. However the use of this training changes with increased experience.

MBAs begin with a bag of tools, e.g. inventory control methods, linear programming, discounted present value, regression forecasting. Their initial impact on the company is made as they find applications for these tools. However, they are often surprised to find that the scope of these methods is less than they anticipated. Their frustration leads them to examine the additional factors not included in their initial models. They begin to take account of certain other variables, e.g. product integrity, customer satisfaction, corporate image, which may not be quantifiable but may be of dominating importance. The influence of the analytic discipline of their early training remains but comes to be applied over more subjective variables. With more experience, analytic habits combine with subjective heuristics to form more refined modes of managerial cognition that Golde (1976) calls 'muddling through'. Muddling through sounds sloppy to logicians, but it does have the important advantage that it tends to account for the right variables (quantifiable or not), and applies them to the right problems. This is contrasted with what Simon (1960/1977:59) calls the 'mathematicians aphasia':

> The victim abstracts the original problem until the mathematical or computational intractibilities have been removed (and all semblance of reality lost), solves the new simplified problem, and then pretends that this was the problem he wanted to solve all along. He hopes the manager will be so dazzled by the beauty of the mathematical formulation that he will not remember that his practical operating problem has not been handled.

Muddling through shares many of the characteristics of mature expertise described by Dreyfus. However, it has even less reliance on its origins in formal, rule-based training. The influence of that training may still remain, especially for some more structured problems, but the content of that training tends to fade away as the manager advances beyond the operational levels. Indeed, most of the great managers of history had no formal managerial training at all (Chandler, 1977).

3. The Hermeneutic Conjecture

An important part of the content of L_{RW} relates to human behavior. In chapter [1], the three levels of management activity described by Anthony (1965) emphasized this aspect in both the management control and strategic planning levels. Management control, concerned with coordinating the various functional activities, deals not with production tasks but rather with the operational managers directing those tasks. Strategic planning is concerned with competitive behavior, product markets, financial markets, general economic and political trends, all of which are fundamentally behavioral.

In all of these areas, the predictive value of formal models is notoriously weak. Yet companies survive and prosper without them. It would appear that human managers have certain skills in these areas that formal methods do not capture.

We all have strong subjective evidence of this on a personal level in our knowledge of peoples' personalities and our ability to empathize. Getting to know a person, we build an internal model of their personality. Depending on the closeness of the acquaintance, we learn their sense of humor, what things will make them angry, and so on. From a formal standpoint, these predictions are incredible. Often we make surprisingly accurate forecasts on apparently sparse information about the individual in question — e.g. a person's politics based on their haircut or clothes. These predictions are of course made with a great deal of background knowledge, e.g. about clothing fashions, cultural norms, political issues.

However, if we consider the complexity of this background knowledge, questions arise as to how were we able to learn it in the first place and how are we able to use it so effectively? Similar types of social skills were also available to the ancient Greeks and, no doubt, before. Yet knowledge about natural phenomena was then much more primitive. What is it that enables us to gauge personalities, the mood at a party or the current political climate, while such other phenomena such as weather, chemical reactions and disease have taken millenia to understand and predict?

Furthermore, our subjective ability to predict behavior seems to collapse almost entirely when it is experienced indirectly, especially when conveyed along a limited number of measurement dimensions. A prospective employee's resume, for instance, is a very weak substitute for a job interview. One may read about a foreign country, but to get a 'feel' for the culture requires that one live there and know the people. President Carter felt the need to make random helicopter visits to various households to get a feel for the current political mood.

The conjecture here, sometimes called *hermeneutics** is that we have the additional information of our own internal experiences (emotions, moods) as material by which to model the personality aspects of others, i.e., to empathize.

Partial support for this conjecture might be found in the success of the Stanislavski method of acting. Here the actor is encouraged not simply to quote his/her lines with certain intonations but rather to

* see for example Gadamer (1976), Apel (1979), Habermas (1981), Ferguson (1981). Klein and Hirschheim (1982) apply hermeneutics to the analysis of office processes.

empathize, indeed to become the character. Our models of other personalities are perhaps analogous to this, though less refined and without the performance aspects.

Hermeneutics, as a theoretical area, is mainly speculative at this point. It does however raise the interesting conjecture that a great deal of managerial cognition may depend on comparisons to internal subjective sensations —information that is not available to formal models.

4. The Ego Problem

People have preferences, computers don't. Computers (as we know them) will never prefer chocolate to vanilla. By preference we mean basic or intrinsic values, as opposed to instrumental or intermediate goals. Chess programs, for instance, have intermediate goals leading to the winning of the game. The goal of winning itself, however, is presumed prior to the system design.

The argument here is not absolute, but rather political. We could for instance imagine a robot with high priority heuristics for survival. This might lead down eventually to a sub-goals such as a taste for sweets or a compulsion to win at chess. However, we aren't likely to *allow* such machines to indulge these preferences if they compete with our own. (Note how Asimov's robots (1978) are ***programmed*** to be socially inferior.) Robot suffrage is not forthcoming.

The converse concept to the social right to have and indulge one's preferences is ***responsibility.*** The outcome of a computer fraud trial is never to put the computer in jail. Interestingly, not only people but also organizations are granted this social status. A corporation (as well, a sovereign state) has independent legal responsibility; it can sign contracts, can be sued, etc.

The preferences (goals, values) of an organization are generally regarded as deriving from the preferences of individuals. Capitalist economics assumes these to be the values of investors. Socialist economics presumes these are imposed by the society at large. Theories of organization, however, tend to ascribe a larger role to the preferences of people within the organization. Cyert and March (1963) note that the influence of stockholders in large corporations has come to be minimal, and regard the preferences of managers as more significant in a predictive theory. Earlier, bureaucracies were characterized as organizations where the influence of individual preferences was minimized. Managers fill prescribed roles and are substitutable over time. The organization's life is not limited to the life of its members. On the other hand, the mechanistic character of bureaucracy, which gives it permanence, also fixes its value structure. Hence railroads, post offices and the military continue to pursue ends that no longer coincide with social interests (Boulding 1978).

In the other extreme, March and Olsen (1979) discuss the nature of organizations where the goals expressed in the organization's formal charter are vague and difficult to measure — e.g. universities, research institutions, charity organizations. Here the organization's goals are heavily influenced by those of individual members, and shift in a fluid way in what they call a 'garbage can process'.

Deal and Kennedy (1982) provide an interesting intermediate viewpoint in their concept of 'corporate culture' (see also Peters 1980). In numerous case examples, for instance IBM, General Electric, Dupont, and 'Japan, Inc.', they observe coordinated, cohesive behavior yet without heavy bureaucratic regulation. The differentiating variable, they argue, is that these organizations have built a strong organizational culture, which influences and molds individual drives and interests to coincide with the organization at large. Conversely, individual preferences and values also exert influence on those of the organization. The dual membership of the individual in the corporate culture as well as the culture at large ensures that the organization maintains goals and values compatible with its larger social context.

The point is that individual preferences play an important role in the adaptation and goodness-of-fit of the organization to its social environment. While we might conceive of a scenario where a robot or information system also displayed intrinsic preferences, this would be socially inadmissible (and has been in all the science fiction to date). It is of course not the preference itself but the tendency to indulge that preference that matters. Having the right to indulge one's preferences (within socially defined bounds) amounts to political participation, a right still not won by all human beings, let alone robots.

We observed in the beginning of this section that an important function of managers is planning. Planning is also an important AI topic. However, one limitation of AI systems to do organizational planning is in the selection of the ultimate preferences and values to which the plans are directed. Another limitation, a semantic one, is discussed next.

D. CONSEQUENCES FOR MANAGEMENT SUPPORT SYSTEMS

This chapter began by postulating a certain triangular relationship, with the descriptive, fact-oriented focus of database management (DM) at one corner, the inferential orientations of artificial intelligence and operations research (AI/OR) at another corner, and the semantic/epistemological focus of decision support systems (DSS) at the third corner.

Moreover, the concerns of DSS are not merely adjuncts to these other disciplines, but involve fundamental problems for information technology applications in general.

Two main issues have been stressed throughout. One is the semantic aspects of data, which were related to the broader issues of formal language semantics. Here the principle concern was the validation of computational inference and the problems posed by semantic change. This led to the second major issue, namely the similarities and differences between human and mechanical types of cognition.

Our conjecture here has been that managerial and machine cognition are in many ways complementary. A management support system ought to enhance that complementarity, but first we need to understand it better.

The use of artificial intelligence for psychological modeling provides a useful input, particularly if we note where the computer models encounter the most difficulty.

A problem here is that we need a way of discussing the two forms of cognition (manager vs machine) in a neutral way. Artificial intelligence is rich with psychological metaphors intended to blur the differences. For management support systems we want to enunciate them.

It is for this reason that a linguistic approach was adopted throughout this book. Computers and people are both language processors. On the other hand, the distinction between formal and natural languages is a long standing one, and underlies much of the modern work in linguistic philosophy.* The argument for the complementarity of managerial vs mechanical cognition is in fact similar to the one Orwell (1963) made in his *1984*. The timing of these concerns is certainly appropriate.

An underlying concern of information technology, particularly knowledge representation, is epistemology. The way we structure knowledge is a matter of pragmatic choice. There are many 'Ways of Worldmaking' (Goodman, 1978). The way we structure our perceptions of the world is the most fundamental of cultural artifacts. Information technology, in this view, is a branch of anthropology, or to use Simon's (1969/1981) term, 'sciences of the artificial'. They are artificial in that the objects they study are the result of human *decision* rather than mere *observation* as in the natural sciences. This suggests a much more general interpretation to the term *decision* support. Decision entails goals, and goals, in turn, entail values (Churchman, 1979). Epistemology interfaces with ethics.

---

* The paradigm example of this is the shift in philosophical position made by Wittgenstein between his early *Tractatus* (1921), and his later *Philosophical Investigations* (1953/1958).

Many technologists have felt an occasional twinge of guilt about the lack of humanism in their work. Most of us manage to shrug it off by an appeal to Adam Smith's invisible hand, or by saying that technology, like fire, is ethically neutral and it's up to sociologists and political scientists to assess its proper role in the society. The problem is that there is very little as regards a conceptual or theoretical link between 'hard' technology and the 'soft' social sciences. The paradigms seem more to compete than cooperate.

Thus one agenda in this book has been to initiate a conceptual bridge between the two by building outwards from the technology side. Our strategy was to work towards the intermediating domains of logic and formal philosophy. We have attempted to employ observations and arguments that are intuitive yet rigorous in an effort to convince the hard-nosed technologists. An important theme has been the theoretical similarity between information technology (which we love) and bureaucracies (which we hate).

Another recurring theme has been the foundational role of contractual relationships both between and within organizations. Contracts are the threads that tie the organization's past to its future. The theory of contracts, however, depends heavily on theories of norms (deontic logics) thus making at least a tenuous link to the study of ethics and value systems.

The overall thrust has been an attempt to bring an epistemological focus to the study of management technology. We are soon to have more technologies than we know what to do with. It is easy to be amazed and dazzled. But the usefulness of these technologies will depend on what they do. What we have them do and not do is the problem of decision support systems.

APPENDIX:

A FORMAL DESCRIPTION OF CONTRACTUAL COMMITMENT

CONTENTS

1. INTRODUCTION

Throughout the text, the fundamental importance of contractual obligation and deontic relationships was stressed as the formal basis for many administrative concepts. The reasoning underlying financial accounting, contract law and bureaucratic regulations is to be found in the eventual extensions of deontic logic. This appendix presents a formal language for describing concepts involving contractual commitment. The language is defined using a model theoretic semantics based on "possible worlds," an approach currently popular in the literature of formal logic and linguistics (see for example, van Fraassen (1971), Thomason (1974), Cresswell (1973)). The notation and form of presentation adopted here is based on (Dowty 1981), which serves as an excellent background tutorial.

The applications of a computer system implementing an axiomatized form of such a formal language are manifold. For instance, much of the legislation regarding contracts, exchange and taxation could potentially be formalized in a language of this sort. Thus, legal retrieval systems such as LEXIS and WESTLAW, which are based on keyword matches, could be superceded by a system performing deductions on theorems expressing the content of the pertinent laws. More than simple retrieval, such a system would be capable of certain analyses that presently require the expertise of a professional lawyer. Even more important, the formulation of laws and regulations in a formal language such as proposed here would allow the system of legislation to be mechanically verified for consistency, completeness, redundancy, etc.

Such a facility could therefore help to remedy a problem, cited in Lee (1983a), with large governmental bureaucracies that make, interpret or enforce these laws: the system of rules becomes much too complex for a single person to comprehend totally. Hence knowledge of the law tends to be spread between multiple individuals, so that use of the laws must contend with the coordination problems between control procedures, paper work, etc. Modification of the laws becomes all the more difficult since it involves not only the legislation itself but also these organizational coordination problems.

The most promising applications, however, would probably arise from the development of a 'logic of bureaucracy', based on deontic logic. The complexity of the rule systems in large bureaucracies is often overhelming. As discussed in the text, this may be the internal rules of an organization; the rule systems of regulatory agencies; or the interlocking rule systems of inter-state and international exchange.

The complexity of these systems becomes so intimidating that they become the private domain of specialists such as lawyers and bureaucrats. However, even the specialists often have lost the sense of effectively managing these rule systems. Rather they simply follow them in the accretion of more and more rules and details.

The generation of natural language "legalese" from the formalized versions of laws and bureaucratic regulations does not present difficult computational problems. A system called AUTOTEXT, written by the author, performed a similar function in a different subject domain (see Lee 1980b, appendix). Going the other way, i.e., converting formal language translations of their natural language forms, however, presents a more formidable problem. In a criticism of certain efforts to use formal languages as a tool for analyzing natural languages, Jardine (1975:229) comments:

> The illusion that much has been achieved in this field may arise from the relative ease with which NL [natural language] sentences can often be generated from sentences of a formal language. But whilst this *may* be a valuable first step towards the construction of rules which "go the other way," in itself it merely corroborates the uncontroversial claim that NL can capture fragments of many formal languages.
>
> To see the gulf which lies between translation from a formal language into NL and its converse, consider definite pronouns. To generate pleasingly colloquial NL representatives for sentences of a predicate calculus it is fairly easy to write programs which eliminate or reduce repetition of names and definite descriptions by introducing definite pronouns, and which do so without introducing unacceptable ambiguities. But "going the other way" it is exceedingly difficult to write a program which disambiguates the reference of definite pronouns using contextual information to find the admissible substitutions of names and definite descriptions.

The applications we foresee for the type of work here, however, avoid this criticism. We do not claim that this formalized language has all the flexibility and nuances capable in natural language. However, the fact that a formal language does not have this flexibility is, we argue, advantageous for these types of applications. One principle difference between a formal and a natural language is that in the first case the rules of interpretation and inference are fixed, whereas in the second they depend on the consensus of the speakers, which may and often does change, even within the span of a single conversation.* In situations of

* Wittgenstein ([ref], [ref]) gives overwhelming evidence of this tendency in ordinary discourse and cites it as the source of many philosophical puzzles. For instance, (1958:9), a diviner might say that when he holds the rod he *feels* that water is five feet under the ground. Here the normal usage of 'feel' is extended. Or, through the influence of AI, we have come to associate the verb 'think' with the behavior of machines. This too is a change of usage that leads us to certain confusions. Wittgenstein (1958:16) suggests that the question "Can a machine think?" is less puzzling when compared to the question "Can a machine have

legislation and regulation this is precisely the feature of natural language that one wants to avoid: the interpretation of these pronouncements should be as fixed and uniform as possible. A way of accomplishing this is to formulate these pronouncements in formal terminology that reduces the dimensions of ambiguity to a limited number of primitive terms.

The purpose of this work is thus one of "explication," Carnap's term for the task of "making more exact a vague or not quite exact concept used in everyday life or in an earlier stage of scientific or logical development" (Carnap 1947). This is regarded as preliminary and complementary to the eventual axiomatization and automation of these concepts.

The language described here is a subset of the notation, called CANDID, originally proposed in Lee (1980a), and extended in Lee (1981). Here we will be primarily concerned with the so-called "deontic"* aspects of that notation. The approach builds on the deontic logic of von Wright (1968). Section II is therefore a summary of von Wright's formalism and its model theoretic interpretation. Section III adds several extensions to this formalism that adapt von Wright's general concepts of obligation, etc., to specific situations of contractual commitment.

The mode of presentation here uses a so-called "model theoretic semantics" (also called "denotational semantics").

Briefly, the idea behind this is that there is some universe of discourse consisting of sets of *objects*. The symbols of the formal language "stand for" or denote these objects. Lisewise, combinations of symbols also have an exact denotation. Thus the syntactic rules describe

---

a toothache?".

* "deontic" refers to concepts of ethical/legal obligation, permission, and prohibition.

the vocabulary of symbols and their allowable combinations while the semantic rules describe the denotation of these individual symbols and their combinatons. One particularly important set in the universe is the set {True, False}, called the set of truth values. Other sets will be added to the universe as we proceed.

II. SUMMARY OF VON WRIGHT'S DEONTIC LOGIC

A deontic logic is one that formalizes the concepts of obligation permission, obligation and prohibition. It is now generally recognized that these concepts are inter-definable — that obligation and permission are logical duals whereas prohibition is the negation of permission.

Von Wright actually presents two deontic calculi, the second being a generalization of the first. Both of these are based on a logic of action, which in turn includes a concept of change.

Our summary will proceed from elementary to complex — i.e., from an ordinary propositional calculus of states, to a calculus of change, and then action, through a modal calculus to the deontic calculi.

A. Propositional Calculus

The various stages of von Wright's deontic logic build on an elementary propositional calculus (PC): By way of introduction, and to help orient the reader to the model theoretic descriptions used throughout this paper, we present this here as the language PC.

1. Syntax of PC

a. Basic expressions

Propositional constants are denoted as single upper case letters or as an alphanumeric string of characters beginning with a capital letter, e.g., P, Q, Raining.

Metalanguage variables for propositions will be denoted as lower case Greek letters, e.g., α, β, γ, Φ, Ψ.

b. Formation rules

The set of ***meaningful expressions***, denoted ME, is defined recursively as follows:

Syn_{PC}.1: Every propositional constant is in ME.

Syn_{PC}.2: If $\Phi \in$ ME then $\sim\Phi \in$ ME.

Syn_{PC}.3 If Φ and Ψ are in ME then so is $(\Phi \,\&\, \Psi)$.

2. Semantic Rules

A model M for PC is any ordered pair <D,F> such that D (the ***universe of discourse***) is a non-empty set of propositional constants and F (the ***interpretation function***) is any function whose domain is D and whose range is the set {False,True}, representing falsehood and truth, respectively. The semantic rules of PC define recursively for any meaningful expression Φ, the ***extension*** of Φ with respect to model M, abbreviated $Den_M(\Phi)$ as follows:

Sem_{PC}.1.: If Φ is any basic expression, the $\text{Den}_M (\Phi) = F(\Phi)$.

Sem_{PC}.2: If $\Phi \in$ ME then $\text{Den}_M\ \Phi$ = True iff $\text{Den}_M \sim\Phi$ is False, and $\text{Den}_M \sim\Phi$ is False otherwise.

Sem_{PC}.3: If Φ and Ψ are in ME, then $\text{Den}_M (\Phi \ \& \ \Psi)$ is True iff both $\text{Den}_M\ \Phi$ and $\text{Den}_M\ \Psi$ are True.

3. Further Definitions

The symbol "::=" is a metalanguage symbol read "is defined as."

For α and β in ME

$$(\alpha \vee \beta) ::= \sim(\sim\alpha \ \& \sim\beta)$$

$$(\alpha \rightarrow \beta) ::= (\sim\alpha \vee \beta)$$

$$(\alpha \leftrightarrow \beta) ::= (\alpha \rightarrow \beta) \ \& \ (\beta \rightarrow \alpha)$$

4. Comment: Logic Proofs in PC

So far, we have described the formal language PC, which gives precise rules for interpreting, i.e., determining the extension or denotation of, any meaningful expression.

As discussed more fully in for example, van Fraasen (1971), a *logic* is a further specification of a formal language that in addition to the above language description also specifies certain expressions in ME as *axioms* and provides certain *transformation* or *inference* rules, which, when applied repeatedly to the axioms, are capable of generating any other meaningful expression in the language. The sequence of transformations that leads to a particular expression is called a *proof* and an expression derived in this way is called a *theorem*. The axioms of a logic are therefore theorems by virtue of a null transformation.

A logic for the propositional calculus language described above is as follows.

a. ***Axioms:*** (from van Fraassen 1971:78)

Ax_{PC}.1: $\alpha \rightarrow (\alpha \,\&\, \alpha)$

Ax_{PC}.2: $(\alpha \,\&\, \beta) \rightarrow \alpha$

Ax_{PC}.3: $\sim(\alpha \,\&\, \beta) \rightarrow \sim(\beta \,\&\, \alpha)$

Ax_{PC}.4: $(\alpha \rightarrow \beta) \rightarrow (\sim(\gamma \,\&\, \beta) \rightarrow \sim(\gamma \,\&\, \alpha))$

b. ***Inference rules:***

IR_{PC}.1: substitution: any meaningful expression may be substituted for the metalanguage variables.

IR_{PC}.2: detachment (modus ponens): if α and $\alpha \rightarrow \beta$, then β

IR_{PC}.3: extensionality if $\alpha \leftrightarrow \beta$, then α may be substituted for β and vice versa, without changing the denotation of the expression in which it appears.

B. Formal Description of Change: The T Calculus

Von Wright interprets the meaningful expressions in PC as representing "some arbitrary state of affairs, such as that it is raining or that a certain window is shut" (von Wright 1968:13). That is, they represent some property of the (actual or possible) world, unbound with respect to time. (This interpretation is discussed in more detail later.)

The first step in extending the PC is to introduce a concept of *change* in these states of affairs. Von Wright does this by introducing a connective T, where Φ T Ψ is read Φ "and then" Ψ. For instance, if R is the proposition "it is raining" and S is the proposition "the sun is shining," then R T S indicates that "it is raining and then the sun is shining." The language for the T calculus (TC) is described as follows:

1. *Syntax*

a. *Basic expressions*

(as for PC)

b. *Formation rules*

The set of meaningful expressions, ME, is defined recursively as follows:

$Syn_{TC}.1$: Every propositional constant is in ME.

$Syn_{TC}.2$: If $\Phi \in$ ME then $\sim\Phi$ is in ME

$Syn_{TC}.3$: If Φ and Ψ are in ME then so is $(\Phi \& \Psi)$

$Syn_{TC}.4$: If Φ and Ψ are in ME then so is $(\Phi \text{ T } \Psi)$.

2. Semantic rules

A model M for TC is any ordered quadruple <D, J, <, F>, where D is a non-empty set of propositional constants, J is a set of points in time ordered by the predicate <, and F is any function whose domain is <D, J> and whose range is the set {False,True}.

The semantic rules of TC define recursively for any meaningful expression Φ, the ***denotation*** of Φ, abbreviated $Den_{M,j}\ \Phi$, as follows:

Sem$_{TC}$.1: If Φ os any basic expression, then $Den_{M,j}\ (\Phi) = F(\Phi,j)$

Sem$_{TC}$.2: If $\Phi \in ME$ then $Den_{M,j}\ {\sim}\Phi$ = True iff $Den_{M,j}\ \Phi$ is False, and $Den_{M,j}\ {\sim}\Phi$ is False otherwise.

Sem$_{TC}$.3: If Φ and Ψ are in ME, then $Den_{M,j}\ (\Phi\ \&\ \Psi)$ is True iff both $Den_{M,j}\ \Phi$ and $Den_{M,j}\ \Psi$ are True.

Sem$_{TC}$.4: If Φ and Ψ are in ME, then $Den_{M,j}\ (\Phi\ T\ \Psi)$ is True iff $Den_{M,j'}\ \Psi$ is True for the unique j' such that for all j'', not $(j < j'' < j')$.

3. Further definitions

(same as for PC).

4. Logic for the T Calculus

Using the axioms and inference rules for the PC logic, von Wright proposes the following additional axioms for the T calculus:

Ax$_{TC}$.1: Distributivity:

$$(\alpha \vee \beta)\ T\ (\Phi \vee \Psi) \leftrightarrow (\alpha\ T\ \Phi) \vee (\alpha\ T\ \Psi) \vee (\beta\ T\ \Phi) \vee (\beta\ T\ \Psi)$$

$Ax_{TC}.2$: Coordination:

$(\alpha \; T \; \beta) \; \& \; (\alpha \; T \; \Psi) \rightarrow \alpha \; T \; (\beta \; \& \; \Psi)$

earlier (von Wright 1965) this was

$(\alpha \; T \; \beta) \; \& \; (\Phi \; T \; \Psi) \leftrightarrow (\alpha \; \& \; \Phi) \; T \; (\beta \; \& \; \Psi)$

$Ax_{TC}.3$: Redundancy:

$\alpha \leftrightarrow \alpha \; T \; (\beta \vee \sim\beta)$

$Ax_{TC}.4$: Impossibility

$\sim(\alpha \; T \; (\beta \; \& \; \sim\beta))$

5. *Additional Theorems, Comments*

$Th_{TC}.1$: $(\alpha \; T \; \beta) \vee (\alpha \; T \sim\beta) \vee (\sim\alpha \; T \; \beta) \vee (\sim\alpha \; T \sim\beta)$

$Th_{TC}.2$: $(\alpha \; T \; \alpha) \vee (\alpha \; T \sim\alpha) \vee (\sim\alpha \; T \; \alpha) \vee (\sim\alpha \; T \sim\alpha)$

This is a corollary of $Th_{TC}.1$. The four disjuncts here are regarded as the four types of elementary changes or state transformations.

$Th_{TC}.3$: $\sim(\alpha \; \& \sim\alpha) \; T \; \beta$

The second Principle of Impossibility.

$Th_{TC}.4$: $(\alpha \; T \; \beta) \rightarrow \alpha$

$Th_{TC}.5$: $\alpha \; \& \; (\beta \; T \; \gamma) \leftrightarrow (\alpha \; \& \; \beta) \; T \; \gamma$

$Th_{TC}.6$: $((\alpha \; T \; \beta) \; T \; \gamma) \leftrightarrow (\alpha \; T \; (\beta \; \& \; \gamma))$

Comment: As indicated by $Th_{TC}.4$ and $Th_{TC}.6$, the perspective of the T connective is from the time of the left argument (em i.e., the right argument is asserted as a state that ***will follow***, but is yet in the future.

Comment (VW): "The connective T is *not* associative. (α T β) T γ is not equivalent to α T (β T γ). The first expression refers, in fact, to *two* successive points in time only, the second refers to *three*."

Comment: This is because (α T β) "resolves to" the time-reference of its first argument. The preceding remark points out that T expressions may be iterated, e.g., α T β T Φ T Ψ. However, because T is not associative this would be syntactically ambiguous. We therefore adopt the convention of evaluation from right to left, e.g.,

$$\alpha \text{ T } \beta \text{ T } \Phi \text{ T } \Psi ::= (\alpha \text{ T } (\beta \text{ T } (\Phi \text{ T } \Psi)))$$

C. Formal Description of Action: The TI Calculus

Von Wright portrays *action* as a composite concept. This depends on another connective, "I" for "instead of," which behaves similarly to T. Indeed, the axioms he proposes that govern I are exactly analogous to those for T. Von Wright (1967:124-5) comments:

> The description to the left of I is, in the I-expression, asserted to hold true of a world in which there is a certain agent. The description to the right holds true of the world which would be, if from the world which is we remove (in thought) the agent.
>
> This "experiment of thought" calls for some comments. The "removal" of the agent does not mean the removal (in thought) of him *body*. The physical presence of the agent may have a causal influence on the world which is not at all connected with his actions. His physical absence would then make a difference to the world, — but *this* difference does not tell us anything about his actions. The "removal" of the agent is the removal (in thought) of whatever *intentions* he may have. It is, therefore, the removal of him *qua agent*.
>
> One could substitute for this experiment of thought one in which the contrast is between a world in which the agent is present physically and a world from which he is about physically. Then the comparison of the states would tell us for which changes and non-changes the agent, through his presence, is *causally*

> *responsible*. This class of changes (and not-changes) includes, but is not necessarily included in, the class of changes (and not-changes) for which he is responsible also *qua agent*.

In von Wright (1968:44-45), he adds:

> Both connectives, "T" and "I", could be called "co-ordinators of possible worlds." "T" coordinates the world which *is* now and the world which *will be* next. "I" coordinates the world *as it is* with an agent in it and the world *as it would be*, if the agent remained passive.

An *action*, indicating the effect of some agent to change the world, involves the combination of a T expression and an I expression in what is called a TI expression:

$$\alpha \, T \, (\beta \, I \, \gamma)$$

is read "α and next β instead of γ," i.e., that because of the influence of some (unspecified) agent, the world changes from state of affairs α to β instead of γ, as it would have without the agent.

Since the connective I really only has interest when combined with T in TI expressions, we skip over a separate description of the "I calculus," and go directly to a statement of the language for the TI calculus, TIC. We see that a new dimension is introduced at this level, that of the application of a proposition not only to a point in time, but also to one or another "possible worlds." At the moment we will assume this to be understood without further explanation. The concept of a possible world will be examined in more detail later on.

THE LANGUAGE TIC:

1. Syntax

a. Basic expressions

(as for PC)

b. Formation rules

The set of meaningful expressions, ME, is defined recursively as follows:

Syn_{TIC}.1: Every propositional constant is in ME.

Syn_{TIC}.2: If $\Phi \in$ ME then $\sim\Phi$ is in ME.

Syn_{TIC}.3: If Φ and Ψ are in ME then so is:

$$(\Phi \,\&\, \Psi)$$
$$(\Phi \; T \; \Psi)$$
$$(\Phi \; I \; \Psi)$$

2. Semantic Rules

A model M for TC is any ordered sextuple, $\langle D, I, Ins, J, <, F \rangle$, where D is a non-empty set of propositional constants, I is a set of possible worlds, Ins is a two place relation coordinating possible worlds, J is a set of times, $<$ is a linear ordering on J, and F is any function whose domain is $\langle D, I, J \rangle$ and whose range is the set {False,True}.

The semantic rules of TIC define recursively for any meaningful expression Φ, the denotation of Φ, abbreviated $Den_{M,i,j}\ \Phi$, as follows:

Sem_{TIC}.1: If Φ is any basic expression, then $Den_{M,i,j}$ Φ = F(Φ,i,j)

Sem_{TIC}.2: If Φ ∈ ME then $Den_{M,i,j}$ ~Φ = True iff $Den_{M,i,j}$ Φ is False, otherwise $Den_{M,i,j}$ ~Φ = False.

Sem_{TIC}.3: If Φ and Ψ are in ME, then $Den_{M,i,j}$ (Φ & Ψ) is True iff both $Den_{M,i,j}$ Φ and $Den_{M,i,j}$ Ψ are True.

Sem_{TIC}.4: If Φ and Ψ are in ME, then $Den_{M,i,j}$ (Φ T Ψ) is True iff $Den_{M,i,j}$ Φ is True and $Den_{M,j,j'}$ Ψ is True for the unique j' such that for all j'', not (j < j'' < j').

Sem_{TIC}.5: If Φ and Ψ are in ME, then $Den_{M,i,j}$ (Φ I Ψ) is in ME iff $Den_{M,i,j}$ Φ is True and $Den_{M,i',j'}$ is True for some world i', such that <j, j'> ∈ Ins and for all times, j'.

3. Further Definitions

(same as PC).

4. Logic for the TI Calculus

Using the inference rules and axioms for the PC logic, as well as the axioms for the TC logic, additional axioms are provided here that control the I connective. As can be seen, they parallel those for T.

For all α, β, Φ, and Ψ in ME:

Ax_{TIC}.1: $(\alpha \vee \beta)\ I\ (\Phi \vee \Psi) \leftrightarrow (\alpha\ I\ \Phi) \vee (\alpha\ I\ \Psi) \vee (\beta\ I\ \Phi) \vee (\beta\ I\ \Psi)$

Ax_{TIC}.2: $(\alpha\ I\ \beta)\ \&\ (\alpha\ I\ \Phi) \rightarrow \alpha I(\beta\ \&\ \Phi)$

Ax_{TIC}.3: $\alpha \leftrightarrow \alpha \, I \, (\beta \vee {\sim}\beta)$

Ax_{TIC}.4: ${\sim}(\alpha \, I \, (\beta \,\&\, {\sim}\beta))$

D. Modals and the Deontic Calculus

von Wright introduces the formal concepts of permission and obligation by extension from interpretations of modal logic.

In modal logic, the notation "$\Diamond\Phi$" is commonly used to indicate "it is possible that Φ." In the terms and to describe the semantics of TIC this would have the interpretation: If $\Phi \in$ ME then $Den_{M,i,j}$ ($\Diamond\Phi$) is True iff $Den_{M,i',j'}(\Phi)$ is True for some i' $\in$ I and some j' $\in$ J.

That is, $\Diamond\Phi$ is true if and only if Φ is true in some possible world at some time. The dual concept of possibility, necessity, is denoted $\Box\Phi$ and is defined as follows:

$$\Box\Phi ::= {\sim}\Diamond {\sim}\Phi$$

These two operators refer to *logical* possibility and necessity. That is, $\Box\Phi$ indicates Φ to be tautological, ${\sim}\Diamond\Phi$ indicates that Φ is contradictory. Between these two is the notion of contingent truth, indicated by $\Diamond\Phi$.

Within this area of logically contingent truth, one can apply the prevailing physical theories and designate certain logically contingent truths to be impossible or necessary according to the laws of nature. If we designate the quality of a world being naturally possible by "Nat," we can then define this more restricted concept of natural possibility, ($\Diamond_N \Phi$) as: If $\Phi \in$ ME then $Den_{M,i,j}$ $\Diamond_N \Phi$ is True iff $Den_{M,i',j'}$ (Φ) is True for some j' $\in$ I such that Nat(j'), and some j' $\in$ J.

The concepts of permission and obligation are developed in analogous fashion. Here, instead of qualifying contingent truth with possibility according to natural laws, it is qualified by its acceptability under some code of ethics or legal system. For the applications we have in mind, this will be the system of laws of some sovereign government (or perhaps a world governing body). The quality of a world being permissible in this system will be designated as "Per." The corresponding concept of deontic possibility might thus be denoted as "$\Diamond_D\ \Phi$." However, following von Wright, we will use the more suggestive notation, $P\Phi$, to indicate that "Φ is permitted."

Its semantic interpretation would then be as follows: If $\Phi \in ME$ then $Den_{M,i,j}\ (P\Phi)$ is True iff $Den_{M,i',j'}\ (\Phi)$ is True for some i' $\in$ I and some j' $\in$ J such that Per(i').

The concept of obligation or deontic necessity, abbreviated "O", is defined as the logical dual:

$$O\Phi ::= {\sim}P \,{\sim}\Phi.$$

Following the semantic definition, this says that Φ must be true in all permitted worlds at all times.

Natural possibility, we observed, was a restriction of the concept of logical possibility. Correspondingly, deontic possibility is reasonably viewed as a restriction on natural possibility. Von Wright (1967:133-4), notes (using "M" for "$\Diamond_N$"):

The concept of possibility within the limits of natural law (including the laws of "human nature") we have denoted by "M". The concept of possibility within the limits of a normative order we shall denote by "P." It seems plausible to regard "P" as the narrower concept in the sense that the expression "P(—)" entails the expression "M(—)," when the blanks in both expressions are filled by the same description of an action or a life. To accept this relation between 'P' and 'M' is tantamount to accepting a (rather strong) version of the well-known principle which is usually formulated in the words "ought implies can."

The language of the deontic calculus, DC, can now be summarized:

1. Syntax of DC

a. Basic expressions

(same as for PC)

b. Formation rules

Same as for TIC with the addition:

Syn_{DC}.4: If Φ is in ME then $P\Phi$ is in ME.

2. Semantic Rules

A model M for DC is any septuple <D, I, Ins, Per, J, <, F>, where D is a non-empty set of propositional constants, I is a set of possible worlds, Ins is a two place relation coordinating possible worlds, Per is a subset of I (the permissible worlds), J is a set of times, < is a linear ordering on J, and F is any function whose domain is <D, I, J> and whose range is the set {False,True}.

The semantic rules of DC define recursively for any meaningful expression Φ, the extension of Φ, denoted $Den_{M,i,j}\ \Phi$ as follows:

Sem_{DC}1-5: (Correspond to semantic rules 1-5 for TIC)

Sem_{DC}.6: If Φ is in ME then $Den_{M,i,j}\ P\Phi$ = True iff $Den_{M,i',j'}\ \Phi$ = True for some $i' \in$ Per and some j'.

3. *Additional Definitions*

Same as for PC with the addition:

$$O\ \Phi ::= {\sim}P\ {\sim}\Phi.$$

4. *Logic for the Deontic Calculus*

a. *Inference rules*

(Same as for PC).

b. *Axioms*

The axioms of PC.

The 4 axioms for T (presented for TC).

The 4 axioms for I (presented for TIC).

Plus:

Ax_{DC}.1: $P(\Phi \vee \Psi) \leftrightarrow P\Phi \vee P\Psi$

Ax_{DC}.2: $P\Phi \vee P\ {\sim}\Phi$

III. SEMANTIC INTERPRETATION

It is important to note how von Wright intends the variables in his calculi to be interpreted. In von Wright (1965:294): the variables (and, presumably their truth functional compounds) refer to "generic propositions" which "are not true or false 'in themselves.' They have a truth-value only relative to a (point in) time. They may be true of one time, false of another. And they may be repeatedly true and false. Let the generic proposition be, e.g., that it is raining. It may be true of today, false of tomorrow, but true again of the day after tomorrow. (The relativity of generic propositions to a location in space will not be considered.)"

In von Wright (1967) he comments:

> The notion of a state of affairs is thus basic to the notion of change. I shall not attempt to answer here the question what a state (of affairs) is. I shall confine myself to the following observation:
>
> One can distinguish between states of affairs in a *generic* and an *individual* sense. Individually the same state, e.g., that the sun is shining in Pittsburgh on 18 March 1966 at 10 a.m., obtains only once in the history of the world. Genercally the same state, e.g., that the sun is shining, can obtain repeatedly and in different places. Of the two senses, the generic seems to me to be the primary one. An individual state is, so to speak, a generic state instantiated ("incarnated") on a certain occasion in space and time.
>
> In the sequel "state" will always be understood in the generic sense. As schematic descriptions of generic states we shall use the symbols $p,q,r,\ldots$, or such letters with an index-numeral.
>
> Let us assume that the total state of the world on a given occasion can be completely described by indicating for every one of a finite number n of states $p_1, \ldots, p_n$ whether it obtains or does not obtain on that occasion. A description of this kind is called a *state- description*. As is well known, the number of possible total states is 2^n if the number of ("elementary") states is n. We can arrange them in a sequence and refer to them by means of state-descriptions: $s_1, \ldots, s_{2n}$.
>
> A world which satisfies the above assumption could be called a *Wittgenstein- world*. It is the kind of world which Wittgenstein

> envisaged in the *Tractatus*. I shall not here discuss the (important) ontological question, whether our real world *is* a Wittgenstein-world, or not. The answer is perhaps negative. But nobody would deny, I think, that, as a simplified model of "a world," Wittgenstein's idea is of great theoretical interest—and state-descriptions of great practical importance. Our study of changes and actions will throughout employ this model.

In a reply to a critique of this paper, von Wright adds:

> I agree with Robison that the distinction between generic and individual states of affairs is problematic. An individual state is spatio-temporally fully specified. A generic state can be generic in the spatial and individual in the temporal component; or *vice versa;* or it can be generic in both components. A description of the total state of the world must, of course, not contain both p and not-p. Therefore, if we let "the world" embrace the whole of space, any generic state of affairs p, the presence or absence of which may be a characteristic of the world, must be individualized in the spatial component. p could then be, e.g., the state that *it is raining in Pittsburgh*. If, on the other hand, we confine "the world" to a specified location ("point") in space, the states of affairs which characterize it need not be individualized in either component. p could now be, e.g., the state that *it is raining*.

In von Wright (1968:13) he starts with the simple explanation: "Let next 'p' represent some arbitrary state of affairs, such as that it is raining or that a certain window is shut." Later, p. 16, he adds:

> A few words should be said about the reading of the formulae. In my first construction of a system of deontic logic the variables were treated as schematic names of actions. According to this conception, "P_p could be read "It is permitted to p." This conception, however, is connected with difficulties and inconveniences. It is, first of all, not clear whether the use of truth-connectives for forming compound names of action is logically legitimate. It is, furthermore, obvious that, on this view of the variables, higher order expressions become senseless. "P_p" itself cannot be the name of an action; therefore it cannot occur within the scope of another deontic operator either.
>
> It now seems to me better to treat the variables as schematic sentences which express propositions. This agrees with the course "taken by most subsequent authors on deontic logic. Instead of "proposition" we can also say "possible state of affairs."—According to this conception, "P_p" may be read "it is

permitted that it is the case that) p."

Against this reading, however, it may be objected that it does not accord very well with ordinary usage. Only seldom do we say of a state of affairs that it is permitted, obligatory, or forbidden. Usually we say this of actions. But it is plausible to think that, when an action is permitted, etc., then a certain state of affairs is, in a "secondary" sense permitted, etc., too. This is the state which, in a technical sense to be explained later, can be called the result of the action in question.

We can take account of this combination of action and resulting state of affairs in our reading of deontic formulae. Instead of saying simply "to p" or "that p" we employ the phrase "see to it that p". "The formulae "P_p" is thus read "it is permitted to see to it that (it is the case that) p" or "one may see to it that p." It should be noted, however, that this reading, though convenient and natural, is somewhat restrictive since it applies *only* to norms which are rules of action.

On p. 18 he adds the additional definitions:

The single variables will be said to represent *elementary states* within the universe. The 2^n different (order of conjuncts being irrelevant) so-called *state- descriptions* in terms of the n variables represent *total states* of the universe. These total states will also be called *possible worlds* (in the universe of elementary states represented by the propositional variables of the set).

As these excerpts illustrate, von Wright uses two kinds of variables (depending on his purposes), an (elementary) *state* (denoted as p,q etc. as in the preceding syntax), and a composite notion that he variously calls a *state description*, *total state*, *Wittgenstein world*, or *possible world*. We belabor this in order to enunciate a change we propose to make in this interpretation.

Von Wright's notion of a possible world seems similar to one that Cresswell (1973:3-4) attributes to Carnap:

Carnap recognizes his debt to Wittgenstein for the notion of a possible world and introduces the notion of a *state- description*. If we assume that there are a set of atomic sentences which may be either true or false without prejudice to the truth or falsity of

> any other atomic sentences then a state-description is a class which contains for every atomic sentence either that sentence or its negation.

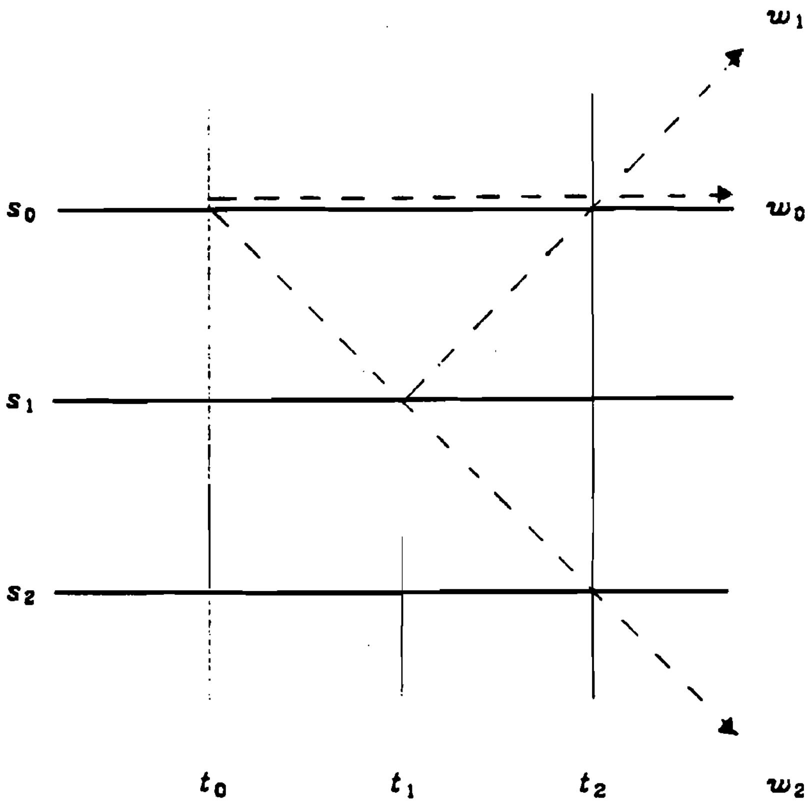
possible world semantics. Cresswell observes (p.4):

> The big advance in the semantical study of modal logic after Carnap was *to remove possible worlds from the dependence on language* which they have in Carnap's work and treat them as primitive entities in their own right, in terms of which the semantical notions required by the modal system can be defined.

In the remainder of this paper we too adopt the view of a possible world as a primitive concept. This view may be related to that of von Wright by means of an intermediate interpretation. Let us refer to von Wright's concept of a possible world as a "VW world" and the more current view of a possible world, as reported by Cresswell, as a "C-world." Let us call the view of a possible world by a third, intermediate interpretation on "I world."

Recall that a VW world was unbound with respect to time. An I world will be a VW world extended across time. An I world is thus *individuated* by a state description at a particular point in time. An I world is therefore by this interpretation a sequence of state description/time point pairs. This is illustrated in Figure 1. s_0, s_1, and s_2 indicate state descriptions, t_0, t_1, and t_2 indicate time points and w_0, w_1, and w_2 indicate possible worlds.

The possible worlds are therefore the paths through these states across time, e.g.,



$w_0 = \{<s_0,t_0>,<s_0,t_1>,<s_0,t_2>\}$

$w_1 = \{<s_0,t_0>,<s_1,t_1>,<s_0,t_2>\}$

$w_2 = \{<s_0,t_0>,<s_1,t_1>,<s_2,t_2>\}$

There are in total 27 such paths, hence 27 possible worlds distinguishable from these three state descriptions and three time points. In general, for m state descriptions and n points in time there will be m^n I worlds (i.e., one choses from m possible states at each of n points in time).

If time is considered to be continuous, the set of I worlds obviously becomes infinite over any interval of time.

Under this interpretation, von Wright's state descriptions become predicates of possible worlds, predicates that uniquely identify an I world at a given time.

The difference between an I-world and a C-world is in the linguistic dependence of the former. In an I-world, a state description, a conjunct consisting of each elementary proposition or its negation, serves to uniquely identify the I world at a point in time. A C-world does not have this feature. For a given state description and point in time, there may be many C-worlds that the vocabulary is not refined enough to distinguish.

In the discussion to follow, we will interpret possible worlds to be C-worlds, unless otherwise indicated.

We now proceed to re-interpret von Wright's operators and connectives according to this view.

IV. EXTENSIONS FOR DESCRIBING CONTRACTUAL COMMITMENT

A contractual commitment (as we view it) differs from the general concept of obligation in that it is an obligation for some particular party, say x, to another party, say y, to do some action, e.g., Φ, within some specified time interval, e.g., before time t. This requires that we bring variables and constants for individual entities and times into the object language.

A. First Order Predicate Calculus

Let us consider first the problem of recognizing entities within the object language. This involves, essentially, extending the role played by the propositional calculus, to that of a first order predicate calculus (FOPC), i.e., introducing individual constants and variables as well as quantifiers.

Partly to set the stage for later developments, we will introduce the FOPC as a "type theoretic" language (see e.g., Dowty (1978: 40-55)). Basically, this approach assigns a syntactic category, called a type, to each of the symbols in the language, and then proceeds to describe further characteristics of the language in terms of relationships between these types. Principally, this allows greater compactness in the language specification.

At this level, there are two basic types, e (for entity) and t (for truth value). Individual constants and variables will have type e, propositions have type t. More complex symbols will be denoted as relations between types. To make effective use of the notation of functional application, these will be confined to two place relations, which may however have other relations in either of their places. So, for instance,

<e,t> is a one place predicate (mapping entities to truth values)

<e,<e,t>> is a two place predicate (mapping entities to one place predicates).

<t,t> is an operator (mapping truth values to truth values)

<t,<t,t>> is a connective (mapping a truth value to an operator).

With this brief background, we introduce the language FOPC.

1. Syntax of FOPC

1 The set of ***types***, defined as follows:

a) e is a type

b) t is a type

c) if a and b are any types, then <a,b> is a type.

2. The basic expressions of FOPC consist of:

constants for each type a

-- constants of type e are denoted as a lower case alpha numeric string beginning with a "@", e.g., @a, @ron, @alec

-- constants of type t or <a,t> where a is any type, are denoted by an alphanumeric string beginning with a capital letter, e.g., P, Q, Raining, Married.

-- all other constants will be assigned special notations in the syntactic rules and definitions.

Variables for each type a.

- variables of type e are denoted as a lower case alpha numeric string beginning with a letter, e.g., x, y, z1, z2.
- variables for all other types are denoted as an alpha numeric string, beginning with a "?", e.g., ?P, ?Q.

Note: in the metalanguage, the italicized letters u and v will be used to denote variables, and as before, lower case Greek letters denote constants.

a. Formation rules of FOPC

The set of *meaningful expressions* of type a, denoted ME_a, for any type a (i.e., the well formed expressions for each type) is defined recursively as follows:

Syn_{FOPC}.1: For each type a, every variable and constant of type a is in ME_a.

Syn_{FOPC}.2: For any types a and b, if $\alpha \in ME_{<a,b>}$ and $\beta \in ME_a$, then $\alpha(\beta) \in ME_b$

Syn_{FOPC}.3: If $\Phi \in ME_t$ and u is a variable (of any type) then $\forall u\ \Phi \in ME_t$

Syn_{FOPC}.4: If $\Phi \in ME_t$ then $\sim\Phi \in ME_t$

Syn_{FOPC}.5: If Φ and Ψ are in ME_t, then $[\Phi\ \&\ \Psi] \in ME_t$

2. Semantics of FOPC

Given a non-empty set D (regarded as the domain of *individuals* or *entities)*, the set of *possible denotations* of meaningful expressions of type a, abbreviated D_a, is given by the following recursive definition:

(1) $D_e = D$

(2) $D_t = \{False, True\}$

(3) $D_{<a,b>} = D_b^{D_a}$ for any types a and b, where Y^X stands for "the set of all possible functions from the set X into the set Y."

A *model* for FOPC is an ordered pair <D,F> such that D is as above and F is a function assigning a denotation to each constant of FOPC of type a from the set D_a.

An *assignment of values to variables* (or simply a *variable assignment)*, g is a function assigning to each variable a denotation from the set D_a for each type a.

The denotation of an expression α relative to a model M and variable assignment g, abbreviated $Den_{M,g}(\alpha)$ is defined recursively as follows:

Sem_{FOPC}.1: If x is a constant, then $Den_{M,g}(\alpha) = F(\alpha)$.

Sem_{FOPC}.2: If x is a variable, then $Den_{M,g}(\alpha) = g(\alpha)$.

Sem_{FOPC}.3: If $\alpha \in ME_{<a,b>}$ and $\beta \in ME_a$, then $Den_{M,g}(\alpha(\beta)) = Den_{M,g}(\alpha)(Den_{M,g}(\beta))$ where Y(X) stands for "the value of the function Y when applied to the argument X."

Sem_{FOPC}.4: If $\Phi \in ME_t$, then $Den_{M,g}(\sim\Phi)$ is True iff $Den_{M,g}(\Phi)$ is False, and $Den_{M,g}(\sim\Phi)$ is False otherwise.

Sem_{FOPC}.5: If Φ and Ψ are in ME_t, then $Den_{M,g}$ $[\Phi \; \& \; \Psi]$ is True iff both $Den_{M,g}$ (Φ) and $Den_{M,g}$ (Ψ) are True.

Sem_{FOPC}.6: If $\Phi \in ME_t$ and u is a variable, then $Den_{M,g}$ $(\forall u \; \Phi)$ = True iff for all g' such that g' is exactly like g except possibly for the value assigned to u, $Den_{M,g'}$ (Φ) = True.

3. *Further Definitions*

For α and β in ME_t:

$$[\alpha \vee \beta] ::= {\sim}[{\sim}\alpha \; \& \; {\sim}\beta]$$

$$[\alpha \rightarrow \beta] ::= [{\sim}\alpha \vee \beta]$$

$$[\alpha \leftrightarrow \beta] ::= [\alpha \rightarrow \beta] \; \& \; [\alpha \rightarrow \beta]$$

For $\Phi \in ME_t$ and u a variable

$$[\exists u \; \Phi] ::= [{\sim}\forall u \, {\sim} \, \Phi]$$

B. Lambda Abstraction

One additional concept will be useful, that of so-called *lambda abstraction*. Dowty (1978:55) introduces this by comparison to the familiar notation for defining a set by means of a predicate, e.g., if Φ is a one place predicate,

$$\{x \mid \Phi(x)\}$$

is the set of individuals in the domain that satisfy this predicate. The operator λ, is used in the object language to the same effect, e.g.,

$$\lambda x\ \Phi x$$

denotes the set of individuals in the domain that satisfy Φ. More specifically, if u is of type e, and e, and $\Phi \in ME_t$, then $\lambda u[\Phi u]$ is the set of <e,t> pairs mapping individuals to truth values.

The converse concept to lambda abstraction is called ***lambda conversion***, which is essentially only functional application. For example, for a variable v, of type e,

$$\lambda u\ [\Phi\ u]\ (v)$$

applies the variable v to the function $\lambda u[\Phi u]$, resulting in $\Phi(v)$. This seems to take us back where we started from in the first place. The advantage however (as Dowty points out) is to make the syntax of the language "flexible." More to the point, it allows reference to predicates and other functions as extensional sets, independent of the variables to which they are applied. (More extensive explanation is given in Dowty, (1978:Section 1.8), and Cresswell, (1973:chapter 6).)

The use of lambda abstraction is not limited to variables of type e, but in fact may be used with variables of any type. Syntactically, it behaves just like the quantifiers, serving to bind the variables.

Recognition of lambda abstraction and conversion in the calculus requires the following additional syntactic and semantic rules:

Syn_λ.1: If $\alpha \in ME_a$ and u is a variable of type b, then $\lambda u\ \alpha \in ME_{<b,a>}$.

Syn_λ.2: If $\alpha \in ME_{<a,b>}$ and $\beta \in ME_a$, then $\alpha\ (\beta) \in ME_b$.

$Sem_\lambda.1$: If $\alpha \in ME_{<a,b>}$ and u is a variable of type b, then $Ext_{M,i,j,g}$ $(\lambda\ u\ \alpha)$ is that function h with domain D_b such that for any object x in that domain $h(x) = Ext_{M,i,j,g'}$ (α), where g' is that value assignment exactly like g with the possible difference that g'(u) is the object x.

$Sem_\lambda.2$: If $\alpha \in ME_{<a,b>}$ and $\beta \in ME_a$, then $Ext_{M,i,j,g}$ $(\alpha\ (\beta))$ is $Ext_{M,i,j,g}$ (α) $(Ext_{M,i,j,g}\ (\beta))$ (i.e., the result of applying the function $Ext_{M,i,j,g}$ (α) to the argument $Ext_{M,i,j,g}$ (β)).

We should note that the introduction of lambda abstraction by comparison to definition of sets by some critical predicate can be slightly misleading. For u a variable of type a, and Φ a predicate,

$$\{u \mid \Phi\ u\}$$

is a set of individuals of type *a*, i.e., the subset of all individuals of type *a* that satisfy Φ.

$$\lambda\ u\ \Phi$$

on the other hand is a set of ordered pairs, <a,t> one for each element of type a in the domain, and whose second place is True if this individual satisfies Φ, False otherwise.

Further, it is seen that the basic information contained in these two concepts is equivalent. Correspondingly, the predicate of elementhood,

$$u \in \alpha$$

has its analog in lambda conversion (functional application):

$$\alpha(u).$$

C. First Order Deontic Calculus

If we now combine this definition of the FOPC language with the extensions von Wright added to the PC, we arrive at a first order deontic calculus, FODC. Its description would be as follows:

1. Syntax of FODC

a. Basic expressions

(same as for FOPC)

b. Formation rules

Syn_{FODC}.1-5: Same as Syn_{PC}.1-Syn_{PC}.5.

Syn_{FODC}.6;7: Same as Syn_{λ}.1, Syn_{λ}2.

Syn_{FODC}.8-9: If Φ and Ψ are in ME_t, then so are

Syn_{FODC}.8: $[\Phi\ T\ \Psi]$

Syn_{FODC}.9: $[\Phi\ I\ \Psi]$

Syn_{FODC}.10: If $\Phi \in ME_t$, then so is $[P\ \Phi]$.

2. Semantics of FODC

Given a non-empty set D (the domain of entities), the set of possible denotations of meaningful expressions of type a, abbreviated D_a, is given by the following recursive definition:

(1) $D_e = D$

(2) $D_t = \{False, True\}$

(3) $D_{<a,b>} = D_b^{D_a}$ for any types a and b.

A ***model*** for FODC is an ordered septuple <D, I, Ins, Per, J, <, F> where D is as above, I is a set of possible worlds, Ins is a two place relation on I coordinating possible worlds (those with and those without the influence, Per is a subset of I (the permissible worlds), J is a set of times, < is a linear ordering on J and F is a function that assigns an appropriate denotation to each constant of FOPC relative to each pair <i,j> for i ∈ I and j ∈ J. (Thus "F(α,<i,j>) = β" is to be interpreted as that the extension (denotation) of α in possible world i at time j is the object β.)

The set of ***possible denotations*** of type a is defined as follows:

$$D_e = D$$

$$D_j = J$$

$$D_t = \{False, True\}$$

$$D_{<a,b>} = D_b^{D_a} \text{ for any types a and b.}$$

A variable assignment, g, is a function assigning to each variable a denotation from the set D_a for each type a.

The denotation of an expression α relative to a model M, a possible world i, time j and value assignment g, abbreviated $Ext_{M,i,j,g}(\alpha)$, is defined recursively as follows:

Sem_{FODC}.1: If α is a constant, then $Ext_{M,i,j,g}(\alpha) = F(\alpha)$.

Sem_{FODC}.2: If α is a variable, then $Ext_{M,i,j,g}(\alpha) = g(\alpha)$.

Sem_{FODC}.3: If $\alpha \in ME_{<a,b>}$ and $\beta \in ME_a$, then $Ext_{M,i,j,g}(\alpha(\beta)) = Ext_{M,i,j,g}(\alpha)(Ext_{M,i,j,g}(\beta))$.

Sem_{FODC}.4: If $\Phi \in ME_t$, then $Ext_{M,i,j,g}(\sim\Phi)$ is True iff $Ext_{M,i,j,g}(\Phi)$ is False and $Ext_{M,i,j,g}(\sim\Phi)$ is False otherwise.

Sem_{FODC}.5: If Φ and Ψ are in ME_t, then $Ext_{M,i,j,g}[\Phi \,\&\, \Psi]$ is True iff both $Ext_{M,i,j,g}(\Phi)$ and $Ext_{M,i,j,g}(\Psi)$ are True.

Sem_{FODC}.6: If $\Phi \in ME_t$ and u is a variable, then $Ext_{M,i,j,g}(\forall u\, \Phi)$ = True iff for all g' such that g' is exactly like g except possibly for the value assigned to u, $Ext_{M,i,j,g'}(\Phi) = 1$.

Sem_{FODC}.7: If Φ and Ψ are in ME_t, then $Ext_{M,i,j,g}[\Phi \; T \; \Psi]$ is True iff $Ext_{M,i,j,g}(\Phi)$ is True and $Ext_{M,i,j',g}(\Psi)$ is True for the unique j' such that for all j'', not $(j < j'' < j')$.

Sem_{FODC}.8: If Φ and Ψ are in ME_t, then $Ext_{M,i,j,g}[\Phi \; I \; \Psi]$ is True iff $Ext_{M,i,j,g}(\Phi)$ is True and $Ext_{M,i',j',g}(\Psi)$ is True for some world i', such that $\langle i,i' \rangle \in Ins$, and for all times, j'.

Sem_{FODC}.9: If $\Phi \in ME_t$ then $Ext_{M,i,j,g}[P\, \Phi]$ = True iff $Ext_{M,i',j',g}(\Phi)$ = True for some $i' \in Per$ and some j'.

3. *Further Definitions*

For α and β in ME_t

$$[\alpha \vee \beta] ::= {\sim}[{\sim}\alpha \,\&\, {\sim}\beta]$$

$$[\alpha \rightarrow \beta] ::= [{\sim}\alpha \vee \beta]$$

$$[\alpha \leftrightarrow \beta] ::= [\alpha \rightarrow \beta] \,\&\, [\beta \rightarrow \alpha]$$

For $\Phi \in ME_t$ and u a variable,

$$[\exists u\, \Phi] ::= [{\sim}\forall u\, {\sim}\Phi]$$

For $\Phi \in ME_t$

$$[O\, \Phi] ::= [{\sim}P\, {\sim}\Phi]$$

The next problem to be considered is the recognition of times *within* the object language. This can be done relatively easily. Adopting a notation suggested by Rescher and Urguhart (1971), the expression

$$(R\ u)\Phi$$

is read that the formula Φ is "realized" at time u. This can be assimilated into the preceding FODC language by means of the following additions.

Consistent with our earlier metalanguage notation using J as a set of times, with j used to indicate elements of J, we revise the specification of types as follows:

e is a type

j is a type

t is a type

if a and b are types, <a,b> is a type.

Variables and constants of type j and type <j,t> will be denoted in the same fashion as variables and constants of type e.

To the formation rules we add the following:

If $\Phi \in ME_t$ and u is a variable of type j, then $[(R\ u)\ \Phi] \in ME_t$.

The denotations of each type are correspondingly as follows:

$$D_e = D$$

$$D_j = J$$

$$D_t = \{False, True\}$$

$$D_{<a,b>} = D_a^{D_b} \text{ for any types a and b.}$$

The following is added to the semantic rules:

Sem_{FODC}.10: If $\Phi \in ME_t$ and u is a variable of type j, then $Ext_{M,i,j,g}$ $[(R\ u)\ \Phi]$ is True iff $Ext_{M,i,j',g}$ (Φ) = True for all j' = g(u).

Several additional definitions will prove useful.

$$u <= v ::= [u < v] \vee [u = v]$$

$$u > v ::= \sim[u <= v]$$

$$u >= v ::= [u > v] \vee [u = v]$$

D. Time Spans

The variables and constants of type <j,t> denote sets of times. Of special interest are sets of contiguous points in time, i.e., ***time spans***. To designate this, we introduce an additional function, span, defined as follows.

For variables u,v and w of type j,

$$\text{span} ::= \lambda u\ \lambda v\ \lambda w\ [(u <= w)\ \&\ (w <= v)]$$

Note that the variables must be of type j, since the definition depends on "<," a relation only defined over the set J.

Span is thus a function of type <j, <j, <j,t>>>. By applying two (time point) arguments to it, e.g., span (u)(v), the result will be of type <j,t>, i.e., the set of points between u and v (or, strictly, the set of pairs <j,t>, indicating by a 1 in the right hand place which points on the time line are between u and v, inclusive.)

Note that by the application of a third argument, e.g., span (u)(v)(w) the result is of type t, i.e., true iff w is between or equal to u and v.

Further realization operators can be defined as convenient. For instance, for u a variable of type <j,t>, and $\Phi \in ME_t$

$$(RT\ u)\ \Phi ::= \forall v\ u(v) \rightarrow (R\ v)\ \Phi$$

Reading: Φ is "realized throughout" time span u.

$$(RD\ u)\ \Phi ::= \exists v\ u(v)\ \&\ (R\ v)\ \Phi$$

Reading: Φ is "realized during" time span u.

We have at this point extended the deontic calculus to recognize individual entities as well as temporal reference. However, several further problems remain in order to adequately describe contractual commitment.

E. Identifying the Agents of Actions

One issue is that we need to particularize actions to identify the agent involved. This entails adding an additional place to the I connective, i.e., of the form (α Iu β). This will lead to a corresponding revision of the predicate Ins, call it Ins', where

$$\text{Ins}'\ (u,i',i'')$$

indicates that world i' is the case rather than i'' due to the influence of agent u.

This requires replacing the former syntactic and semantic rules for I as follows:

$\text{Syn}'_{\text{FODC}}$.9:If α and β are in ME_t and u is a variable or constant of type e, then (α Iu β) $\in ME_t$.

$\text{Sem}'_{\text{FODC}}$.9:If Φ and Ψ are in ME_t, and u is a variable or constant of type e, $\text{Ext}_{M,i,j,g}$ [Φ Iu Ψ] is True iff $\text{Ext}_{M,i,j,g}$ (Φ) is True and $\text{Ext}_{M,i'',j',g}$ (Ψ) is True for some world i' such that <g(u),<i', i''>> $\in$ Ins', for all times j'.

When substituted in a TI expression this provides an explication for the sense that x *does* some action Φ.

We still however need to account for the sense that x is obligated *to y* to do Φ. Before addressing that, however, we need to introduce a notation for contingent permission and obligation.

F. Contingent Permission and Obligation

Von Wright goes beyond the deontic definitions described so far to what he calls a "dyadic" version of the deontic logic. For various reasons (noted in the appendix), we are unable to incorporate that here. However, we do have need of an analogous concept to his contingent permission and obligation. Using a notation analogous to his, we write

$$P\ \alpha/\beta$$

to indicate that in some permissible world, both β and α are true. Contingent obligation is defined as

$$O\ \alpha/\beta ::= {\sim}P\ {\sim}\alpha/\beta$$

which may be interpreted that in any world, if β is true then if the world is permissible, then α is true.

The scoping and quantification may be a bit hard to follow in these explanations. To help clarify, we will temporarily make use of formal notation in the metalanguage, distinguishing this from the object language by enclosing it in double brackets, e.g., [[ ]].

In this notation, w will be a variable for possible worlds.

Thus,

$$P\ \alpha/\beta ::= [[\exists w\ \beta(w)\ \&\ Per(w)\ \&\ \alpha(w)]]$$

$$O\ \alpha/\beta ::= [[{\sim}\exists w\ \beta(w)\ \&\ Per(w)\ \&\ {\sim}\alpha(w)]]$$

$$\leftrightarrow [[\forall w\ {\sim}\beta(w)\ V\ {\sim}Per(w)\ V\ \alpha(w)]]$$

$$\leftrightarrow [[\forall w\ \beta(w) \rightarrow (Per(w) \rightarrow \alpha(w))]]$$

We find it useful to generalize these concepts of conditional permission and obligation to arbitrarily many levels. We therefore define

$$P(\alpha / \beta_1 / \beta_2 / \ldots / \beta_n) ::=$$
$$[[\exists w\, \beta_n(w) \,\&\, \ldots \,\&\, \beta_2(a) \,\&\, \beta_1(w) \,\&\, Per(w) \,\&\, \alpha(w)]]$$

Analogously, we define the generalized form of conditional obligation as:

$$D(\alpha / \beta_1 / \beta_2 / \ldots / \beta_n) ::= {}^{\sim}P({}^{\sim}\alpha / \beta_1 / \beta_2 / \ldots / \beta_n)$$
$$\leftrightarrow [[{}^{\sim}\exists w(\beta_n(w) \,\&\, \ldots \,\&\, \beta_2(w) \,\&\, \beta_1(w) \,\&\, Per(w) \,\&\, {}^{\sim}\alpha(w))]]$$
$$\leftrightarrow [[\forall w\, {}^{\sim}\beta_n(w) \vee \ldots \vee {}^{\sim}\beta_2(w) \vee {}^{\sim}\beta_1(w) \vee Per(w) \vee {}^{\sim}\alpha(w)]]$$
$$\leftrightarrow [[\forall w\, \beta_n(w) \rightarrow (\ldots \rightarrow (\beta_2(w) \rightarrow (\beta_1(w) \rightarrow (Per(w) \rightarrow \alpha(w)]]]$$

(Here the additional square right bracket is meant to close all open left hand parentheses.)

To incorporate these concepts of conditional permission and obligation in the formal language, the following additions are needed:

Syn. If α, β_2, ..., β_n are all in ME_t, then $P(\alpha / \beta_2 / \ldots / \beta_n)$ is in ME_t.

Sem. If α, β_1,..., β_n are all in ME_t, then $Ext_{M,i,j,g}\, P(\alpha / \beta_1 / \ldots / \beta_n) = 1$ iff for some i', i' $\in$ Per, and $Ext_{M,i',j,g}(\alpha)$ is True and $Ext_{M,i',j,g}(\beta_k)$ is True for $k = 1, \ldots, n$.

Def. If α, β_2, ..., β_n are all in ME_t, then $O(\alpha / \beta_1 / \ldots / \beta_n) ::= {}^{\sim}P({}^{\sim}\alpha / \beta_1 / \ldots / \beta_n)$.

G. **The Benefactors of Contractual Commitments**

As mentioned above, while the formal language is now refined to distinguish the agent of actions in contractual commitments, we yet lack a way of identifying the other party, what we might call the "benefactor" of the obligation or permission.

The commitment to this party might at first examination be considered as a sort of local obligation separate from the overall legal system represented by O and the other deontic operators. However, if when we deal with contractual, as opposed to say informal, obligation between two parties, we are nonetheless referring to obligations allowed and enforced within a broad system of contract law. There are therefore certain circumstances prescribed in law that allow x to become (legally) obligated to g to do Φ.

For instance, x's obligation to give y some object, say z, may only come in force if y pays x some sum of money (perhaps only a partial or token payment). Contracts are thus often stated as pairs of obligations, with opposite roles of the same two parties. However, neither obligation may in fact become effective until all or part of the other has been executed. These conditions for creating a contractual obligation, however, depend on the specifications of the legal system governing the parties.

(International contracts, involving perhaps several legal systems, entail further complications that we ignore here.)

By this view x becomes generally obligated to do Φ. That is however not quite the case in a contractual obligation. In a contract, if y defaults and does not do Φ, y has recourse to certain *legal actions* against x. But these do not come automatically; y must initiate them in the form of a lawsuit, or some similar type of appeal to the governing body for enforcement of his/her claims against x.

This leads us to the view that contractual obligation is not a general obligation for x to do Φ, but rather a permission on the part of y to take legal action against x if x does not do Φ. This notion of "legal action" can obviously be very complex and as well varies depending on the government having jurisdiction. I do believe though that the possibility of taking legal action is a necessary element to explicate obligation. It is therefore adopted as a primitive predicate, namely,

$$LA(x,y)$$

indicates a "legal action of x against y."

With this assumption, we are now able to define a concept of contractual obligation:

$$O(x,y)\ \Phi ::= P\ LA\ (y,x) / {\sim}\Phi(x).$$

$O(x,y)\ \Phi$ has the reading that "x is obligated to y to Φ," and is defined as the permission of y to take legal action against x if x does not Φ.

Note that "O" here for contractual obligation is not the same as the O for general obligation. The two are distinguished by the presence of the two arguments in the case of contractual obligation.

As was the case with general obligation and permission, we take contractual obligation and permission to be dual concepts:

$$P(y,x)\ \Phi ::= {\sim}O(x,y)\ {\sim}\Phi$$
$$::= {\sim}[P\ LA(y,x)/{\sim}({\sim}\Phi(x))]$$
$$::= {\sim}P\ LA(y,x)/\Phi(x).$$

Note that the places are reversed in contractual permission and its dual obligatory form. The definition says that if y permits x to Φ, then y is not permitted to take legal action against x if x does Φ.

This conforms with usual intuitions. A contractual permission of y to x allows x to do something he/she would normally be forbidden (not permitted) to do, i.e.,

$${\sim}P(y,x)\ \Phi ::= O(x,y)\ {\sim}\Phi$$
$$::= {\sim}P\ LA(y,x)/\Phi(x)$$

i.e., normally, y would be allowed legal action against x if x did Φ. A permission to do Φ is thus a suspension of this right to take legal action.

The concepts of conditional obligation and permission can be extended to the contractual case:

$$O(x,y)\ \Phi/\Psi ::= P[LA(y,x)/{\sim}\Phi(x)/\Psi]$$
$$::= [[\exists w\ \Psi(w)\ \&\ {\sim}\Phi(x,w)\ \&\ Per(w)\ \&\ LA(y,x,w)]]$$

Reading: x is obligated to y to do Φ given Ψ is defined that it is permitted for y to take legal action against x given that x does not do Φ given Ψ, which, in the symbolic metalanguage form, is in turn defined that in some permitted world, x has not done Φ, Ψ is true and y takes legal action. Correspondingly,

$$P(x,y)\ \Phi/\Psi ::= {\sim}O(y,x)\ {\sim}\Phi/\Psi$$

$$::= {\sim}P\ LA(x,y)/{\sim}\Phi(y)/\Psi$$

$$::= [[\forall w\ \Psi(w) \rightarrow (\Phi(y,w) \rightarrow (Per(w) \rightarrow LA(x,y,w)))]]$$

Reading: the permission of x to y to do Φ given Ψ is defined (last line) that in any possible world, if Ψ is true then if y does Φ then if the world is permitted there is no legal action taken by x against y.

In all the above cases, the enforcement of the contractual obligation (or permission) has been the application (or suspension) of some 'legal action', which we have adopted as a primitive concept. However, in many contracts, the enforcement is a specific action that we would want to explicate in the calculus, e.g., the right to claim ownership of some particular asset serving as collateral for a loan in the case of default.

We will indicate the relationship to an enforcement action by the connective OE read "or else."

In the case of contractual obligation this is defined:

$$O(x,y)\ \Phi\ OE\ \gamma ::= P(x,y)\ \gamma/{\sim}\Phi(x)$$

Reading: the obligation of x to y to do Φ or else γ is defined as the permission of x to y given that x does not do Φ.

This has a natural extension to cases of conditional contractual obligation:

$$O(x,y)\ \alpha/\beta\ OE\ \gamma ::= P(x,y)\ \gamma/{\sim}\alpha(x)/\beta$$

Reading: the obligation of x to y to do x given β or else γ is defined as the permission of x to y to do γ given that x does not do α given β.

Specific enforcements may likewise be considered for contractual permission, though this is much less natural—(indeed I can think of no practical example).

The definition would go as follows:

$$P(x,y)\ \Phi_y\ OE\ \gamma ::= {\sim}O(x,y)\ {\sim}\Phi_y\ OE\ \gamma ::= {\sim}P(y,x)\ \gamma_x/\Phi(y)$$

Reading: the permission of x to y to Φ or else γ is to say that y is not obligated to x not to Φ or else γ, which is to say that y does not permit x to γ given that y does Φ.

I. Formal Summary: Language CC (Contractual Commitment)

1. Syntax of CC

a. Types

Let t, e and j be any fixed objects. Then the set of *types* is defined recursively as follows:

i. t is a type

ii. e is a type

iii. j is a type

iv. If a and b are types, then <a,b> is a type.

b. Basic expressions

i. For each type a, CC contains a denumerably infinite set of *non-logical constants* (or simply *constants*), $C_{n,a}$, for each natural number n. The set of all constants of type a is denoted Con_a.

ii. For each type a, CC contains a denumerably infinite set of *variables* $V_{n,a}$ for each natural number n. The set of all variables of type a is denoted Var_a.

c. Syntactic rules of CC

The set of *meaningful expressions* of type a, denoted ME_a, is defined recursively as follows:

Syn_{CC}.1: Every variable of type a is in ME_a

Syn_{CC}.2: Every constant of type a is in ME_a

Syn_{CC}.3: If $\alpha \in ME_a$ and u is a variable of type b, then $\lambda u \alpha \in ME_{<b,a>}$.

Syn_{CC}.4: If $\alpha \in ME_{<a,b>}$ and $\beta \in ME_a$, then $\alpha(\beta) \in ME_b$.

Syn_{CC}.5: If α and β are both in ME_a, then $\alpha = \beta \in ME_t$.

Syn_{CC}.6-7: If Φ and Ψ are in ME_t, then the following are also in ME_t:

Syn_{CC}.6: $\sim\Phi$

Syn_{CC}.7: $\Phi \,\&\, \Psi$

Syn_{CC}.8: If $\Phi \in ME_t$ and u is a variable of any type, then $\forall u\ \Phi \in ME_t$

Syn_{CC}.9: If Φ and Ψ are in ME_t, then $\Phi\ T\ \Psi \in ME_t$

Syn_{CC}.10: If Φ and Ψ are in ME_t and u is of type e then $\Phi\ Iu\ \Psi \in ME_t$.

Syn_{CC}.11: If $\Phi \in ME_t$, then $P\Phi \in ME_t$

Syn_{CC}.12: If $\alpha, \beta_1, \ldots, \beta_n$ are all in ME_t, then $P(\alpha / \beta_1 / \ldots / \beta_n) \in ME_t$

Syn_{CC}.13: If $\Phi \in ME_t$ and u is a variable of type j, then $[R\ u\ \Phi] \in ME_t$

2. *Semantics of CC*

A ***model*** for CC is an ordered octuple <D, I, Ins', Per, LA, J, <, F> such that D, I and J are non-empty sets, Ins' is a relation on D X I X J, (where one world is a counter factual alternative to another because of the influence of some agent in D). Per is a subset of I (the permitted worlds). LA is a relation on D X D X I (the predicate for legal action). < is a linear ordering on the set J, and F is a function that assigns an appropriate denotation to each constant of CC relative to each pair <i,j> for $i \in I$ and $J \in J$. The set of ***possible denotations*** of type a is defined as follows:

i. $D_e = D$

ii. $D_j = J$

iii. $D_t = \{False, True\}$

iv. $D_{<a,b>} = D_b^{D_a}$ for any types a and b.

An assignment of values to variables, g, is a function having as domain the set of all variables and giving as value for each variable of type a a member of D_a.

The denotation of an expression α relative to a model M, a possible world i, time j, and value assignment g, abbreviated $Ext_{M,i,j,g}(\alpha)$, is defined recursively as follows:

Sem_{CC}.1: If α is a constant, then $Ext_{M,i,j,g}(\alpha) = F(\alpha)$

Sem_{CC}.2: If α is a variable, then $Ext_{M,i,j,g}(\alpha) = g(\alpha)$.

Sem_{CC}.3: If $\alpha \in ME_{<a,b>}$ and u is a variable of type b, then $Ext_{M,i,j,g}(\lambda u\ \alpha)$ is that function h with domain D_b such that for any object x in that domain, $h(x) = Ext_{M,i,j,g'}(\alpha)$, where g' is that value assignment exactly like g with the possible difference that g'(u) is the object x.

Sem_{CC}.4 If $\alpha \in ME_{<a,b>}$ and $\beta \in ME_a$, then $Ext_{M,i,j,g}(\alpha(\beta))$ is $Ext_{M,i,j,g}(\alpha)(Ext_{M,i,j,g}(\beta))$ (i.e., the result of applying the function $Ext_{M,i,j,g}(\alpha)$ to the argument $Ext_{M,i,j,g}(\beta)$).

Sem_{CC}.5: If α and β are in ME_a, then $Ext_{M,i,j,g}(\alpha = \beta)$ is True if and only if $Ext_{M,i,j,g}(\alpha)$ is the same as $Ext_{M,i,j,g}(\beta)$.

Sem_{CC}.6: If $\Phi \in ME_t$, then $Ext_{M,i,j,g}(\sim\Phi)$ is True if and only if $Ext_{M,i,j,g}(\Phi)$ is False, and $Ext_{M,i,j,g}(\sim\Phi)$ is False otherwise.

Sem_{CC}.7: If Φ and Ψ are in ME_t, then $Ext_{M,i,j,g}[\Phi \& \Psi]$ is True if and only if both $Ext_{M,i,j,g}(\Phi)$ and $Ext_{M,i,j,g}(\Psi)$ are True.

Sem_{CC}.8: If $\Phi \in ME_t$ and u is a variable of type e, then $Ext_{M,i,j,g}(\forall u\ \Phi)$ is True if and only if $Ext_{M,i,j,g}(\Phi)$ is True for all g' exactly like g except possibly for the value assigned to u.

Sem_{CC}.9:. If Φ and Ψ are in ME_t, then $Ext_{M,i,j,g}(\Phi\ T\ \Psi)$ is True iff $Ext_{M,i,j,g}(\Phi)$ is True and $Ext_{M,i,j',g}(\Psi)$ is True for the unique j' such that j < j' and for all j'', either not j < j'' < j' or j'' = j'.

Sem_{CC}.10: If Φ and Ψ are in ME_t and u is of type e then $Ext_{M,i,j,g}[\Phi\ Iu\ \Psi]$ is True iff $Ext_{M,i,j,g}\ \Phi$ is True and $Ext_{M,i',j,g}(\Psi)$ is True for some i' such that <g(u),i,i'> $\in$ Ins'.

Sem$_{CC}$.11: If $\Phi \in ME_t$, then $Ext_{M,i,j,g}\ P\Phi$ is True iff $Ext_{M,i',j',g}\ \Phi$ is True for some i' such that i' $\in$ Per and some j'.

Sem$_{CC}$.12: If $\alpha, \beta_1, \ldots, \beta_n$ are all in ME_t, then $Ext_{M,i,j,g}\ P(\alpha/\beta_1/\ldots/\beta_n)$ is True iff for some i', such that i' $\in$ Per and $Ext_{M,i,j,g}\ (\alpha)$ is True and $Ext_{M,i',j,g}\ (\beta_k)$ is True for $\beta_k = \beta_1/\ldots/\beta_n$.

Sem$_{CC}$.13: If $\Phi \in ME_t$ and u is a variable of type j, then $Ext_{M,i,j,g}\ [R\ u\ \Phi]$ is True iff $Ext_{M,i',j',g}\ (\Phi)$ = True for all j' = g(u).

3. Additional Definitions

i.-iii. For α and β in ME_t

i. $[\alpha \vee \beta] ::= \sim[\sim\alpha \,\&\, \sim\beta]$

ii. $[\alpha \rightarrow \beta] ::= [\sim\alpha \vee \beta]$

iii. $[\alpha \leftrightarrow \beta] ::= [\alpha \rightarrow \beta] \,\&\, [\beta \rightarrow \alpha]$

iv. For $\Phi \in ME_t$ and u and v variables of type e,

$\exists u\, \Phi ::= \sim\forall u \sim \Phi$

v. For $\Phi \in ME_t$ $O\Phi ::= \sim P \sim\Phi$

vi. If $\alpha, \beta_1, \ldots, \beta_n$ are all in ME_t, then $O(\alpha/\beta_2/\ldots/\beta_n)$ $::= \sim P \sim\alpha/\beta_1/\ldots/\beta_n$.

For u, v and w variables of type j,

vii. $[u \geq v] ::= [u < v] \vee [u = v]$

viii. $[u > v] ::= \sim[u \leq v]$

ix. $u \geq v ::= [u > v] \vee [u = v]$

x. $\text{span} ::= \lambda u\ \lambda v\ \lambda w\ [(u \leq w)\ \&\ (w \leq v)]$

For u a variable of type t, and $\Phi \in ME_t$

xi. $RT\ u\ \Phi ::= \forall v\ u(v) \rightarrow [R\ v\ \Phi]$

xii. $RD\ u\ \Phi ::= \exists v\ u(v) \rightarrow [R\ v\ \Phi]$

If $\Phi \in ME_t$ and u and v are of type e, then

xiii. $O(u,v)\ \Phi ::= P\ LA(v,u) / {\sim}\Phi(u)$

xiv. $P(u,v)\Phi ::= {\sim}O(v,u)\ {\sim}\Phi$

If α, β and γ are in ME_t and u and v are variables of type e, then

xv. $O(x,y)\ \alpha/\beta ::= P(LA(v,u)/{\sim}\alpha(x)/\beta)$

xvi. $P(u,v)\ \alpha/\beta ::= {\sim}O(v,u)\ {\sim}\Phi/\Psi$

xvii. $O(u,v)\ \alpha\ OE\ \gamma ::= P(u,v)\ \gamma/{\sim}\alpha(x)$

xviii. $O(u,v)\ \alpha/\beta\ OE\ \gamma ::= P(u,v)\ \gamma/{\sim}\alpha(u)/\beta$

REFERENCES

Abrahamsson, B. 1977. *Bureaucracy or Participation: The Logic of Organization*. London: Sage Publications.

Abrial, J.R. 1974. Data Semantics. In J.W. Klimbie and K.L. Koffeman, eds. *Data Base Management*, pp.1-59. Amsterdam: North-Holland.

Adams, E.W. 1979. Measurement Theory. In P.D. Asquith and H.E. Kyburg, Jr. eds., *Current Research in Philosophy of Science*, pp.207-227. East Lansing: Michigan.

Anthony, R.N. 1965. *Planning and Control Systems: A Framework for Analysis*. Cambridge, Massachusetts: Harvard University Graduate School of Business Administration, Studies in Management Control.

Apel, K-O. 1979. *The Explanation: Understanding Controversy in Transcendental-Pragmatic perspective*. (Title translated from German: Theorie-Diskussion). Frankfurt: Suhrkamp.

Ashton, R.H. 1977. Objectivity of Accounting Measures: A Multirule-Multimeasurer Approach. *The Accounting Review*, LII(3):567-575.

Asimov, I. 1978. *I, Robot*. New York: Fawcett.

Austin, J.L. 1962. *How to Do Things with Words*. Cambridge, Mass.: Harvard University Press.

Barstow, D.R. 1979. *Knowledge- Based Program Construction*. New York: North Holland.

Beaver, W., and Demski, J.S. 1974. The Nature of Financial Accounting Objectives: A Summary and Synthesis. *Journal of Accounting Research*, pp.170-187.

Berlinsky, D. 1976. *On Systems Analysis: An Essay Concerning the Limitations of Some Mathematical Methods in the Social, Political and Biological Sciences*. Cambridge: MIT Press.

Biller, H. and Neuhold, E.J. 1978. Semantics of Data Bases: The Semantics of Data Models. *Information Systems*, 3:11-30.

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

Boulding, K.E. 1956. *The Image*. Ann Arbor: The University of Michigan Press.

Boulding, K.E. 1978. *Stable Peace*. Austin: University of Texas Press.

Brachman, R.J. 1978. A Structural Paradigm for Representing Knowledge. BBN Report No. 3605. Cambridge, Massachusetts: Bolt Beraneck & Newman.

Brachman, R.J. 1979. On the Epistemological Status of Semantic Networks. In N.V. Findler, ed., *Associative Networks: The Representation and Use of Knowledge by Computers*. pp.3-50. New York: Academic Press.

Brachman, R.J. and Smith, B.C., eds. 1980. Special Issue on Knowledge Representation. *SIGART NEWSLETTER* No.70.

Brodie, M.L. 1982. Axiomatic Definitions for Data Model Semantics. *Information Systems*. 7(2):183-197.

Buchanan, B.G. and Feigenbaum, E.A. 1978. Dendral and Meta-Dendral: Their Applications Dimension. *Artificial Intelligence*, 11(1,2):5-24.

Burns, T., and Stalker, G.M. 1961. *The Management of Innovation*. London: Tavistock.

Carlson, M.L., and Lamb, J.W. 1981. Constructing a Theory of Accounting —An Axiomatic Approach. *The Accounting Review*, LVI(3):554-573.

Carnap, R. 1947. *Meaning and Necessity*. Chicago: University of Chicago Press.

Caspari, J.A. 1976. Wherefore Accounting Data — Explanation, Predication and Decisions. *The Accounting Review*, LI(4):739-745.

Castañeda, H-N. 1981. The Paradoxes of Deontic Logic: The Simplest Solution to All of Them in One Fell Swoop. In R. Hilpinen 1981, pp.37-86.

Chambers, R.J. 1955. Blueprint for a Theory of Accounting. *Accounting Research*, January:17-25.

Chambers, R.J. 1966. *Accounting, Evaluation and Economic Behavior*. Englewood Cliffs, New Jersey: Prentice-Hall. Reprint ed., Houston: *Accounting Classics Series.*, Scholars Book Co., 1975.

Chambers, R.J. 1972. Measurement in Current Accounting Practice: A Critique. *The Accounting Review*, July:488-509.

Chambers, R.J. 1973. Accounting Principles or Accounting Policies? *The Journal of Accountancy*, May:48-53.

Chandler, A.D. Jr. 1977. *The Visible Hand.* Cambridge, Massachusetts: The Belknap Press of Harvard University Press.

Chang, C.C. and H.J. Keisler. 1973. *Model Theory*. Volume 73, A. Heyting, *et al.* (eds.) *Studies in Logic and The Foundations of Mathematics*. Amsterdam: North-Holland, and New York: American Elsevier Publishing Co. Inc.

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

Churchman, C.W. 1979. *The Systems Approach and its Enemies*. New York: Basic Books.

Clocksin, W.F. and Mellish, C.S. 1981. *Programming in Prolog*. New York: Springer-Verlag.

Cohen, M.D., March, J.G. and Olsen, J.P.. 1972. A Garbage Can Model of Organizational Choice. *Administrative Science Quarterly*. 17(1):1-25.

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

Codd, E.F. 1972. Further Normalization of the Data Base Relational Model. In R. Rustin, ed., *Data Base Systems* Courant Computer Science Symposia 6. Englewood Cliffs, New Jersey: Prentice-Hall., also IBM Research Report RJ909.

Codd, E.F. 1979. Extending the Database Relational Model to Capture More Meaning. *ACM Transactions on Database Systems*, 44(December):397-434.

Coelho, H., Cotta, J.C. and Pereira, L.M. 1980. *How to Solve It With Prolog*. 2nd Edition. Lisbon: Laboratório Nacional de Engenharia Civil.

Cooper, A.A., and Parker, J.E. 1973. Human Resource Accounting: An Examination. *Cost and Management*, Jan-Feb:21-26.

Craswell, A.T. 1978. Surrogates in Accounting. *ABACUS*, 14(1):81-93.

Cresswell, M.J. 1973. *Logics and Languages*. London: Nethuen & Co. Ltd.

Crozier, M. 1964. *The Bureaucratic Phenomenon*. Chicago: University of Chicago Press.

Cyert, R.M. and March, J.G.. 1963. *A Behavioral Theory of the Firm*. Englewood Cliffs, New Jersey: Prentice-Hall.

Date, C.J. 1977. *An Introduction to Database Systems*, 2nd edition. Reading, Massachusetts: Addison-Wesley.

Davis, R., and J. King. 1975. An Overview of Production Systems, Stanford AI Lab Memo AIM-271, *Stanford Computer Science Report*. STAN-CS-75-524, October.

Davis, R., and Lenat, D.B. 1982. *Knowledge- Based Systems in Artificial Intelligence*. New York: McGraw-Hill International.

DeMillo, R.A., Lipton, R.J. and Perlis, A.J. 1979. Social Processes and Proofs of Theorems and Programs. *Communications of the ACM*, 22(5):271-280.

Deal, T.E. and Kennedy A.A. 1982. *Corporate Cultures*. Reading, Massachusetts: Addison-Wesley.

Diffie, W. and Hellman, M.E. 1979. Privacy and Authentication: An Introduction to Cryptography. *Proceedings of the IEEE*, 67(3):397-427.

Dobrov, G.M. 1983. The Strategy for Organized Technology in the Light of Hard-, Soft-, and Org-ware Interaction. *Long Range Planning*, 12(4), reprinted as RR-83-2. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Downs, A. 1967. *Inside Bureaucracy*. Boston: Little, Brown & Co.

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

Dreyfus, H. 1979. *What Computers Can't Do*. Revised Edition. New York: Harper Colophon Books, Harper and Row.

Dreyfus, S.E. 1982. Formal Models vs. Human Situational Understanding: Inherent Limitations on the Modeling of Business Expertise. *Office Technology and People*, 1(2-3):133-165.

Ellis, B. 1966. *Basic Concepts of Measurement*. Cambridge: Cambridge University Press.

Enderton, H.B. 1972. *A Mathematical Introduction to Logic*. New York: Academic Press.

Everest, G.C., and Weber, R. 1977. A Relational Approach to Accounting Models. *The Accounting Review*, LII(2):340-359.

Fagin, R. Multivalued Dependencies and a New Normal Form for Relational Databases. *ACM Transactions on Database Systems*, 2(3):262-278.

Ferguson, M. 1981. Empathy as an Instrument of Science. Working Paper. Chicago: University of Illinois at Chicago Circle.

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

Fick, G.P. 1980. Small Computers in Organizations: Issues and Arguments — or — How to Fight with Computer-Enhanced Bureaucracy. WP-80-146. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Findler, N.V. ed. 1979. *Associative Networks: The Representation and Use of Knowledge by Computers.* New York: Academic Press.

Flores, F., and J.J. Ludlow. 1980. Doing and Speaking in the Office. In: G. Fick and R.H. Sprague, Jr. eds. *Decision Support Systems: Issues and Challenges.* Oxford: Pergamon.

Fodor, J.A. 1981a. Tom Swift and His Procedural Grandmother. In Fodor, J.A. *Representations*, Chapter 8. pp.205-224. Brighton: The Harvester Press.

Fodor, J.A. 1981b. Methodological Solipsism Considered as a Research Strategy in Cognitive Psychology. In Fodor, J.A. *Representations*, Chapter 9. pp.225-253. Brighton: The Harvester Press.

Fox, M.S. 1979. Organization Structuring: Designing Large Complex Software. CMU-CS-79-155. Pittsburgh: Department of Computer Science, Carnegie-Mellon University.

Føllesdal, D. and Hilpinen, R. 1971. Deontic Logic: An Introduction. In R. Hilpinen, ed., 1971/1981, pp.1-35.

Frankena, W.K. 1967. Value and Valuation. In P. Edwards, ed. *The Encyclopedia of Philosophy*, Vol.8, pp.229-232. New York: Macmillan and the Free Press.

Gadamer, H.G. 1976. *Philosophical Hermeneutics.* Edited and translated by D.e. Linge. Berkeley: University of California Press.

Galbraith, J. 1973. *Designing Complex Organizations.* Reading, Massachusetts: Addison-Wesley.

Galbraith, J. 1977. *Organization Design.* Reading, Massachusetts: Addison-Wesley.

Gallaire, H., Minker, J. and Nicolas, J.M., eds. 1981. *Advances in Data Base Theory* Volume 1. New York and London: Plenum Press.

Gallaire, H., and Minker, J., eds. 1978. *Logic and Data Bases.* New York and London: Plenum Press.

Ghiselli, E.E., Campbell, J.P., and Zedeck, S. 1981. *Measurement Theory for the Behavioral Sciences.* San Francisco: W.H. Freeman.

Ginzberg, M.J., Reitman, W. and Stohr, E.A. 1982. *Decision Support Systems.* Amsterdam: North-Holland.

Golde, R.A. 1976. *Muddling Through: The Art of Properly Unbusinesslike Management*. New York: AMACOM.

Goodman, N. 1951/1977. *The Structure of Appearance*. Third Edition. Dordrecht: D. Reidel.

Goodman, N. 1972. *Problems and projects*. New York: The Bobbs-Merrill.

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

Goodman, N. 1979. *Fact, Fiction, and Forecast*. Indianapolis: Hackett Publishing.

Gorry, G.A. and Scott-Mortonm M.S.. 1971. A Framework for Management Information Systems. *Sloan Management Review*, 13(1):55-70.

Grady, P. 1965. *Inventory of Generally Accepted Accounting Principles for Business Enterprises*. Accounting Research Study No.7. New York: American Institute of CPAs.

Griffith, R.L. Three Principles of Representation for Semantic Networks. *ACM Transactions on Database Systems*, 7(3):417-442.

Habermas, J. 1981. *Theory of Communicative Action*. (Title translated from German: Theory des kommunikativen Handelns). Frankfurt: Suhrkamp.

Hammer, M., and McLeod, D. 1981. Database Description with SDM: A Semantic Database Model. *ACM Transactions on Database Systems*, 6(3):351-386.

Hampton, D.R., Summer, C.E. and Webber, R.A. 1978. *Organizational Behavior and the Practice of Management*. Third Edition. Glenview, Illinois: Scott, Foresman and Company.

Haseman, W.D., and Whinston, A.B. 1976. Design of a Multidimensional Accounting System. *Accounting Review*, 51(1):65-79.

Hilpinen, R. 1971/1981. ed. *Deontic Logic: Introductory and Systematic Readings*. Dordrecht: D. Reidel.

Hilpinen, R. 1981. ed. *New Studies in Deontic Logic*. Dordrecht: D. Reidel.

Hintakka, J. 1975. Carnap's Heritage in Logical Semantics. In Hintakka, J. ed., *Rudolf Carnap, Logical Empiricist*. pp.217-242. Dordrecht: D. Reidel.

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

Hughes, J.S. 1978. Toward A Contract Basis of Valuation in Accounting. *The Accounting Review*, LIII(4):882-894.

Hume, D. 1739. *A Treatise on Human Nature*. London.

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

Ijiri, Y. 1971. Fundamental Queries in Aggregation Theory. *The Journal of the American Statistical Association*, 66(336):766-782.

Ijiri, Y. 1975. *Theory of Accounting Measurement*. Sarasota, Florida: American Accounting Association.

Infotech. 1981. *Machine Intelligence*. Infotech State of the Art Report,

Jacques, E. 1976. *A General Theory of Bureaucracy*. London: Heinemann.

Jardine, N. 1975. Model Theoretic Semantics and Natural Language. In E.L. Keenan, ed. *Formal Semantics of Natural Language*, pp.219-240. Cambridge: Cambridge University Press.

Johnson, O. 1975. Toward an "Events" Theory of Accounting. *The Accounting Review*, April:641-653.

Jones, S., Mason, P., and Stamper, R. 1979. LEGOL 2.0: A Relational Specification Language for Complex Rules. *Information Systems*. 4:293-305.

Joseph, E.C. 1982. What's Ahead for Intelligence Amplification? *FUTURICS* 6(2):35-38.

Kalish, D., Montague, R. and Mar, G.. 1980. *Logic: Techniques of Formal Reasoning*, Second Edition. New York: Harcourt Brace Jovanovich, Inc.

Keen, P.G.W. and Scott-Morton, M.S. 1978. *Decision Support Systems*. Reading, Massachusetts: Addison-Wesley.

Kent, E.W. 1981. *The Brains of Men and Machines*. Peterborough, N.H.: BYTE/McGraw Hill.

Kent, W. 1978. *Data and Reality*. Amsterdam: North-Holland.

Kent, W. 1979. Limitations of Record-Based Information Models. *ACM Transactions on Database Systems*, 4(1):107-131.

Klein, H.K., and Hirschheim, R. 1982. Issues and Approaches to Appraising Technological Change in the Office: A Consequentialist Perspective. WP-8206-1.0. Hamilton, Ontario: ISRAM, McMaster University, Faculty of Business

Kowalski, R. 1979. Algorithm = Logic + Control. *Communications of the ACM*, 22(7):424-436.

Kowalski, R. 1979a. *Logic for Problem Solving*. New York and Oxford:

North Holland.

Krantz, D.H., Luce, R.D., Suppes, P., and Tversky, A. 1971. *Foundations of Measurement*, Volume 1, Additive and Polynomial Representations. New York: Academic Press.

Kripke, S.A. 1963. Semantical Considerations on Modal Logics. *Acta Philosophica Fennica, Modal and Many- Valued Logics* pp.83-94.

Kripke, S. 1971. Identity and Necessity. In M.K. Munitz, ed. *Identity and Individuation.*, pp. 135-164. New York: New York University Press. Also reprinted in: S.P. Schwartz, ed. 1977. *Naming, Necessity, and Natural Kinds*, London: Cornell University Press.

Kripke, S.A. 1972. Naming and Necessity. In Davidson and Harman, eds., *Semantics of Natural Language* pp.253-355.

Kuhn, T.S. 1962. *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press.

Lee, R.M., and Gerritsen, R. 1978. Extended Semantics for Generalization Hierarchies. *Proceedings of the International Conference on management of Data.* Austin, Texas, June, 1978, pp.18-25.

Lee, R.M., and Krigman, A. 1978. Generating Semi-Structured Text: Representation. Working Paper 78-08-03. Philadelphia: Department of. Decision Sciences, The Wharton School, The University of Pennsylvania.

Lee, R.M. 1980a. CANDID: A Logical Calculus for Describing Financial Contracts. Ph.D. dissertation, available as WP-80-06-02. Philadelphia, PA: Department of Decision Sciences, the Wharton School, University of Pennsylvania.

Lee, R.M. 1980b. ICON: A Case Study in Office Automation and Microcomputing. WP-80-183. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Lee, R.M. 1981. 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. The Technological Obsolescence of Mediocrity. WP-82-76. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Lee, R.M. 1983a. Bureaucracies, Bureaucrats and Information Technology. WP-83-27. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Lee, R.M. 1983b. Applications Software and Organizational Change: Issues in the Representation of Knowledge. WP-83-30. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Lee, R.M. 1983c. Analyzing Red Tape: The Performative vs Informative Roles of Bureaucratic Documents. WP-83-39. Laxenburg, Austria: International Institute for Applied Systems Analysis.

Levelt, W.J.M. 1974. *Formal Grammars in Linguistics and Psycholinguistics*. Volume 1. The Hague: Mouton.

Lieberman, A.Z., and Whinston, A.B. 1975. A Structuring of an Events-Accounting Information System. *The Accounting Review*, April:246-258.

March, J.G. and Simon, H. A. 1958. *Organizations*. New York: Wiley.

March, J.G. and Olsen, J.P. 1979. *Ambiguity and Choice in Organizations*. 2. Edition. Bergen: Universitetsforlaget.

Martin, W.A. and Fateman, R.J.. 1971. The MACSYMA System. Proceedings of ACM 2nd Symposium on Symbolic and Algebraic Manipulation. Los Angeles, California. pp.23-25.

Maslow, A.H. 1943. A Theory of Human Motivation. *Psychological Review*. 50(4):370-96.

Mattessich, R. 1957. Towards a General and Axiomatic Foundation of Accountancy — With an Introduction to the Matrix Formulation of Accounting Systems. *Accounting Research*, October:328-355.

Mattessich, R. 1964. *Accounting and Analytical Methods*. Homewood, Illinois: R.D. Irwin. Reprint, Houston, ed., *Accounting and Classics Series*, Scholars Book Co., 1977.

Mattessich, R. 1978. *Instrumental Reasoning and Systems Methodology*. Dordrecht: D. Reidel.

Mattessich, R. 1980. On the Evolution of Theory Construction in Accounting: a Personal Account. *Accounting and Business Research*. Spring:1-16.

McCarthy, W.E. 1979. An Entity-Relationship View of Accounting Models. *The Accounting Review*, October:667-686.

McCarthy, W.E. 1982. The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. *The Accounting Review*, LVII(3):554-578

McCorduck, P. 1979. *Machines Who Think*. San Francisco: W.H. Freeman.

McDermott, D. 1980. The Prolog Phenomenon. *SIGART Newsletter*, 72(July):16-20.

McDermott, D. A Temporal Logic for Reasoning About Processes and Plans. *Cognitive Science*, 6:101-155.

McDonald, D.D. 1977. Language Generation: The Linguistics Component.

Proceedings of the Fifth International Joint Conference on Artificial Intelligence. Cambridge, Massachusetts. p.142.

McGregor, D. 1960. *The Human Side of Enterprise*. New York: McGraw-Hill.

Mellor, D.H. 1981. *Real Time*. Cambridge: Cambridge University Press.

Michael, D.N. 1977. Planning's Challenge to the Systems Approach. In H.A. Linstone and W.H.C. Simmonds, eds. *Futures Research*. Reading, Massachusetts: Addison-Wesley.

Miller, G.A. 1956. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review*, 63(2):81-97.

Minsky, M. 1975. A Framework for Representing Knowledge. In P.H. Winston, ed., *The Psychology of Computer Vision*. pp.211-277. New York: McGraw-Hill.

Mintzberg, H. 1973. *The Nature of Managerial Work*. New York: Harper & Row.

Montague, R. 1973. The Proper Treatment of Quantification in Ordinary English. In Thomason, R.H. ed. 1974, pp. 247-270.

Montague, R. 1974. English as a Formal Language. In Thomason, R.H., 1974, pp. 188-221.

Moonitz, M. 1961. *The Basic Postulates of Accounting*. New York: American Institute of CPAs. Accounting Research Study No.1.

Moto-oka, T. ed. 1981. *Fifth Generation Computer System*, Proceedings of the International Conference on Fifth Generation Computer Systems, Tokyo, Japan, October 19-22, 1981. Amsterdam: North-Holland.

Mylopoulos, J., Bernstein, P.A., and Wong, H.K. 1980. A Language Facility for Designing Database-Intensive Applications. *ACM Transactions on Database Systems*, 5(2):185-207.

Newell, A., and H. Simon. 1972. *Human Problem Solving*. Englewood Cliffs, New Jersey: Prentice Hall.

Newell, A., J. Shaw, and H. Simon. 1963. Empirical Explorations of the Logical Theory Machine. In *Computers and Thought*, E. Feigenbaum and J. Feldman (eds.), New York: McGraw-Hill, pp.109-113.

Nilsson, N.J. 1980. *Principles of Artificial Intelligence*, Palo Alto, CA: Tioga Publishing Co.

Norman, D.A., D.E. Rumelhart, and the LNR Research Group. 1975. *Explorations in Cognition*. San Francisco: Newman.

Orwell, G. 1963. *Nineteen Eighty-Four*. New York: Harcourt-Brace

Jovanovich.

Peters, T.J. 1980. Management Systems: The Language of Organizational Character and Competence. *Organizational Dynamics*. Summer.

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

Putnam, H. 1970. Is Semantics Possible? In: H.E. Keefer and M.K. Munitz, eds, *Language, Belief, and Metaphysics*. New York: State University of New York Press. Also reprinted in: Schwartz, S.P., ed., 1977.

Putnam, H. 1978. *Meaning and the Moral Sciences*. Boston: Routledge & Kegan Paul.

Quillian, M.R. 1968. Semantic Memory. In M. Minsky, ed., *Semantic Information Processing*. Cambridge, Massachusetts: MIT Press.

Quine, W.V.O. 1953/1961. On What There Is. In *From a Logical Point of View. Logico- Philosophical Essays*. Second Edition. New York: Harper & Row, pp. 1-19.

Quine, W.V.O. 1969. Natural Kinds. In *Ontological Relativity and Other Essays*. pp.114-138. New York: Columbia University Press.

Quine, W.V.O. 1974. *The Roots of Reference*. La Salle, Illinois: Open Court.

Quine, W.V.O. *Word and Object*. Cambridge, Massachusetts: The MIT Press.

Rescher, N. 1975. *A Theory of Possibility*. Pittsburgh: University of Pittsburgh Press.

Rescher, N. and A. Urquhart. 1971. *Temporal Logic*. Wien: Springer-Verlag.

Roberts, F.S. 1979. *Measurement Theory with Applications to Decision-making, Utility, and the Social Sciences*. Reading, Massachusetts: Addison-Wesley.

Robinson, R.A. 1965. A Machine Oriented Logic Based on the Resolution Principle. *Journal of the ACM*, 12(January):23-41.

Ross, A. 1941. Imperatives and Logic. *Theoria*, 7:53-71.

Sartre, J.P. 1938/1965. *Nausea*. Middlesex: Penguin Books. (Translated from French, 1938. Paris: Librairie Gallimard.)

Schank, R., and Abelson, R. 1977. *Scripts, Plans, Goals and Understanding*. Hillsdale, New Jersey: Erlbaum Associates.

Schumacher, E.F. 1971. *Small is Beautiful. A Study of Economics As if People Mattered*. London: Blond & Briggs.

Schwartz, S.P. ed. 1977. *Naming, Necessity, and Natural Kinds*, London: Cornell University Press.

Searle, J.R. 1969. *Speech Acts: An Essay in the Philosophy of Language*. London: Cambridge University Press.

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

Sibatani, A. 1980. The Japanese Brain. *Science 80*, December:22-27.

Simon, H.A. 1955. A Behavioral Model of Rational Choice. *The Quarterly Journal of Economics*, Vol. LXIX.

Simon, H.A. 1960/1977. *The New Science of Management Decision*. Englewood Cliffs, New Jersey: Prentice-Hall.

Simon, H.A. 1969/1981. *The Sciences of the Artificial*. Cambridge, MA: The MIT Press.

Simon, H.A. 1981. Studying Human Intelligence by Creating Artificial Intelligence. *American Scientist* 69(May-June):300-309.

Smith, J.M., and Smith, D.C.P. 1977. Database Abstractions: Aggregation and Generalization. *ACM Transactions on Database Systems*, (2) June:105-133.

Sol, H.G. 1983. *Processes an Tools for Decision Support*. Proceedings of IFIP/IIASA Working Conference on Processes and Tools for Decision Support, July 19-21, 1982, Laxenburg, Austria. Amsterdam: North-Holland.

Sombart, W. 1928. *Der moderne Kapitalismus*, 6th ed. München and Leipzig: Duncker und Humbolt.

Sorter, G.H. 1969. An "Events" Approach to Basic Accounting Theory. *The Accounting Review*, January:12-19.

Sproull, L., Weiner, S. and Wolf, D. 1978. *Organizing an Anarchy*. Chicago: University of Chicago Press.

Stamper, R. 1973. *Information in Business and Administrative Systems*. New York: Wiley.

Stamper, R. 1977. Physical Objects, Human Discourse and Formal Systems. In G.M. Nijssen, ed., *Architecture and Models in Data Base Management Systems*. pp.293-311. Amsterdam: North-Holland.

Stevens, S.S. 1959. Measurement Psychophysics, and Utility. In C.W. Churchman and P. Ratoosh, eds., *Measurement: Definitions and Theories*, pp.18-63. New York: Wiley

Strawson, P.F. 1959. *Individuals*. Garden City, New York: Anchor Books.

Suppes, P. 1957. *Introduction to Logic*. New York: D. van Nostrand

Company.

Tarski, A. 1941/1965. *Introduction to Logic and to the Methodology of Deductive Sciences*. New York: Oxford University Press.

Tarski, A. 1956. The Concept of Truth in Formalized Languages. In A. Tarski, *Logic, Semantics, Metamathematics* (translated by J.H. Woodger. Originally presented to the Warsaw Scientific Society, March 1931 in Polish. Oxford: Clarendon Press.

Taylor, F.W. 1911. *The Principles of Scientific Management*. New York: Harper and Row.

Thom, R. 1975. *Structural Stability and Morphogenesis. An Outline of a General Theory of Models*. Reading, Massachussetts: W.A. Benjamin.

Thomason, R.H. ed. 1974. *Formal Philosophy—Selected Papers by Richard Montague*. New Haven: Yale University Press.

Thomason, R.H. 1981. Deontic Logic as Founded on Tense Logic. In R. Hilpinen, ed., 1981, pp.165-176.

Thompson, G.B. 1981. Ethereal Goods, The Economic Atom of the Information Society. Paper prepared for the EEC Conference on the Information Society, Dublin, Ireland, 18-20 November 1981. Ottawa: Bell Northern-Research.

Thompson, V.A. 1961. Bureaucracy and Bureaupathology. In: Hampton, et al. eds. *Organizational Behavior and the Practice of Management*. Glenview, Illinois: Scott Foresman.

Toffler, A. 1980. *The Third Wave*. New York: Bantam Books.

Tsichritzis, D. and Klug, A. eds. 1977. The ANSI/X3/SPARC DBMS Framework. Report of the Study Group on Data Base Management Systems. Montvale, New Jersey: AFIPS Press.

Tversky, A. and Kahneman, D. 1974. Judgment under Uncertainty: Heuristics and Biases. *Science*, 185(September):pp.1124-1131.

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

von Wright, G.H. 1951. Deontic Logic. *Mind* 60:1-15. Reprinted in *Logical Studies* (by G.H. von Wright), London: Routledge and Kegan Paul, 1957, pp.58-74.

von Wright, G.H. 1965. And Next. *Acta Philosophica Fennica*, Fasc. XVIII, Helsinki. pp. 293-301.

von Wright, G.H. 1967. The Logic of Action—A Sketch. in N. Rescher, ed. *The Logic of Decision and Action*. pp.121-136. Pittsburgh: University of Pittsburgh Press.

von Wright, G.H.. 1968. An Essay in Deontic Logic and the General Theory of Action. *Acta Philosophica Fennica*, Fasc. XXI, Amsterdam: North-Holland.

von Wright, G.H. 1971/1981. Deontic Logic and the Theory of Conditions. In R. Hilpinen, ed., 1971/1981, pp.159-177.

von Wright, G.H. 1981. On the Logic of Norms and Actions. In R. Hilpinen, ed., 1981., pp.3-36.

Watts, R.L., and Zimmerman, J.L. 1979. The Demand for and Supply of Accounting Theories: The Market for Excuses. *The Accounting Review*. LIV(2):273-305.

Weber, M.. 1956/1978. *Economy and Society*. Berkeley, California: University of California Press, translated from *Wirtschaft und Gesellschaft*, Tuebingen: J.C.B. Mohr, 1956.

Weizenbaum, J. 1976. *Computer Power and Human Reason*. San Francisco: Freeman.

Whorf, B.L. 1956. *Language, Thought and Reality: Selected Writings of Benjamin Lee Whorf*, edited by J.B. Carroll. New York: Wiley.

Wiener, N. 1967. *Human Use of Human Beings: Cybernetics and Society* New York: Avon Books.

Williamson, O.E. 1973. Markets and Hierarchies: Some Elementary Considerations. *American Economic Review*, 63(2):316-325.

Winograd, T. 1975. Frame Representations and the Declarative/Procedural Controversy. In D.G. Borrow and A. Collins, eds. *Representation and Understanding*, pp.185-210. New York: Academic Press.

Winograd T. 1976. Towards a Procedural Understanding of Semantics. AIM-292 (Report No. STAN-CS-76-580) Stanford, California: Computer Science Department, Stanford University.

Winograd, T. 1979. Beyond Programming Languages. *Communications of the ACM*, 22(7):391-401.

Winston, P.H. 1977. *Artificial Intelligence*. Phillipines: Addison-Wesley.

Wittgenstein, L. 1921. *Tractatus Logico- Philosophicus*. English translation by D.F. Pears & B.F. McGuinness, 1961, London and Henley: Routledge & Kegan Paul.

Wittgenstein, L. 1953/1958. *Philosophical Investigations*. Translated by G.E.M. Anscombe. Third Edition. New York: Macmillan.

Wittgenstein, L. 1958/1965. *The Blue and Brown Books*. New York: Harper & Row.

Woodward, J. 1978. Management and Technology. In D.R. Hampton, C.E. Summer and R.A. Webber, eds, *Organizational Behavior and the*

Practice of Management. pp.344-353. Scott, Foresman & Co.

Wulf, W.A. 1977. Some Thoughts on the Next Generation of Programming Languages. In A.K. Jones, ed. *Perspectives on Computer Science*, New York: Academic Press.

Zadeh, L.A., Fu, K.S., Tanaka, K. and Shimura, M. (eds.). 1975. *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*. New York: Academic Press.