A LOGICAL ANALYSIS OF SOFT SYSTEMS MODELLING: IMPLICATIONS FOR INFORMATION SYSTEM DESIGN AND KNOWLEDGE BASED SYSTEM DESIGN.

FRANK HUTSON GREGORY
Bachelor of Arts (York),
Master of Science (Lancaster).

A thesis submitted for
the Degree of Doctor of Philosophy
to the University of Warwick

November 1993

Warwick Business School,
Faculty of Social Studies,
the University of Warwick.

ACKNOWLEDGMENTS

Thanks are due to David Miller of the Philosophy Department, University of Warwick, for providing the formal proof of the meta-rule given in Appendix 2 (section 11.2). Also to David Lockwood of Warwick Business School, University of Warwick, for advising on the database third normal form given in section 8.5.5.

John Mingers and Yasmin Merali of Warwick Business School made a wide ranging contribution to the development of the thesis by numerous helpful comments and, in the case of the latter, lively debates. Many of the ideas in the thesis originated in discussions with Mr. Mingers or Ms. Merali. As is usually the case it is difficult to determine who came up with an idea first. Where the thesis draws on the written work of Mingers or Merali these are acknowledged in the appropriate place. Naturally the author takes full responsibility for all errors.

As supervisor for the doctoral research Mr. Mingers put in far more hours than is officially allocated for this work and is especially deserving of thanks.

Thanks are also due to Prof. Robert Galliers of Warwick Business School for advice and the general direction of the research.

The findings in this paper were the result of research funded by a studentship provided by the Science and Engineering Research Council (SERC) and a bursary provided by Warwick Business School.

DECLARATION

The thesis draws on a number of papers which the author has previously published. These are:

SSM to Information Systems: A Wittgensteinian Approach. *Journal of Information Systems*, (1993) **3**, pp. 149 -168.

Cause, Effect, Efficiency & Soft Systems Models. *Journal of the Operational Research Society* Vol. 44 (4), pp. 333 - 344. 1993.

SSM for Knowledge Elicitation & Representation. *Proceedings of the Conference on the Theory, Use and Integrative Aspects of IS Methodologies*. British Computer Society Information Systems Methodologies Special Interest Group. Sept. 1993. pp. 377 - 390.

Mapping Conceptual Models on to the Real World. *Proceedings of the United Kingdom Systems Society 3rd International Conference*, Plenum Publishing Corp. July 1993. pp. 117 - 122.

Logic and Meaning in Conceptual Models: Implications for Information System Design. Proceedings of the UKSS Information & Information Systems Seminar. *Systemist* Vol. 15 (1). Feb. 1993. pp. 28 - 43.

SSM to Information Systems: A Logical Account. Proceedings of the UKSS Soft Systems Methodology and Information Systems Seminar, *Systemist* Vol. 14 (3), Aug. 1992. pp. 180 - 189.

SSM for Knowledge Elicitation & Representation. *Warwick Business School Research Paper No. 98.* 1993.

(Frank Gregory & Yasmin Merali) Inductions, Modality & Conceptual Modelling. *Warwick Business School Research Paper No. 79.* 1993.

SSM to Information Systems: A Wittgensteinian Account. *Warwick Business School Research Paper No. 65.* 1992.

Logic & Meaning in Conceptual Models: Implications for Information System Design. *Warwick Business School Research Paper No. 62.* 1992.

Cause, Effect, Efficiency & Soft Systems Models. *Warwick Business School Research Paper No. 42.* 1992.

Causation and Soft Systems Models. *Systemist,* Vol. 13 (3), Aug 1991. pp. 105 - 112.

The preliminary work on the logical analysis of soft systems modelling is contained in:

The Conceptual Basis of Information Requirements Analysis. MSc Dissertation. Department of Systems and Information Management. University of Lancaster. Sept, 1990.

SUMMARY OF THE THESIS

The thesis undertakes an analysis of the modelling methods used in the Soft Systems Methodology (SSM) developed by Peter Checkland and Brian Wilson. The analysis is undertaken using formal logic and work drawn from modern Anglo-American analytical philosophy especially work in the area of philosophical logic, the theory of meaning, epistemology and the philosophy of science.

The ability of SSM models to represent causation is found to be deficient and improved modelling techniques suitable for cause and effect analysis are developed. The notional status of SSM models is explained in terms of Wittgenstein's language game theory. Modal predicate logic is used to solve the problem of mapping notional models on to the real world.

The thesis presents a method for extending SSM modelling in to a system for the design of a knowledge based system. This six stage method comprises: systems analysis, using SSM models; language creation, using logico-linguistic models; knowledge elicitation, using empirical models; knowledge representation, using modal predicate logic; codification, using Prolog; and verification using a type of non-monotonic logic. The resulting system is constructed in such a way that built in inductive hypotheses can be falsified, as in Karl Popper's philosophy of science, by particular facts. As the system can learn what is false it has some artificial intelligence capability. A variant of the method can be used for the design of other types of information system such as a relational database.

PART I

1 INTRODUCTION

1.1 The nature of the inquiry

1.1.1 The nature of the problem

The normal way of presenting a thesis is to give a clear statement of the problem and then proceed to its solution. The procedure is apt when the nature of the problem is conceptually simple and easily recognized. When the problem is highly abstract and conceptually complex this procedure is not possible. With conceptually complex problems demonstrating that there is a problem often involves considerable work, and a clear statement of the problem yet more work. In analytical philosophy, which is concerned with conceptually complex problems, the greater part of an academic paper will usually be devoted to giving a clear statement of the problem. When this is done the solution is often fairly simple.

The thesis will identify a number of minor logical problems that can be found in the modelling techniques used in SSM (Soft Systems Methodology). Once a problem has been identified, recommendations will be made as to how the SSM models can be developed in order to overcome the problem. Solutions will be found by using more powerful logic which results in more powerful models. More powerful models open up a range of applications, such as cause and effect analysis, and knowledge elicitation, not found in traditional SSM.

Underlying the minor logical problems is a fundamental problem. This is a problem not just for SSM but for all methods of systems analysis that can be used for information system design. A precise statement of this problem will require the use of modal logic and modal logic will provide its solution. The greater part of the thesis will be concerned with demonstrating that modal logic is relevant to information system design. Thus the nature of the problem will not be made absolutely clear until the latter stages of the thesis. However, a preliminary statement of the problem can be given in order to orientate the reader.

The fundamental problem is the problem of real world mapping, and this is the problem of how something that people have made up can tell us anything about things that have not been made up. It is a well recognized problem in mathematics. It was expressed rather nicely in a recent book review:

> One of the great mysteries of existence is the way in which mathematics - a human mental construct - captures aspects of reality. Stewart (p. 27, 1993).

Real world mapping is a problem that will arise in the design of any computerized information system. It will be a problem for all information systems be they for business, administration or science. As hardware constraints on computer systems diminish, it is a problem that will rise to the surface. However, this fact is not generally recognized and the vast majority of people who work in this area are unaware of the problem. This is just as true of academics as it is of practitioners.

One of the main objectives of the thesis will be to show that real world mapping is a problem for information system designers. However, a considerable amount of logical scene setting will be required before the problem can be presented in a tractable form. Once this is done a theoretical solution is presented and this in turn suggests a direction in which a practical solution might be found. The thesis expounds a theory of information system design and puts forward the outline of a practical method of information system design simultaneously.

The problem could have been addressed solely at the level of high level theory in which case the thesis would be a work of pure analytical philosophy. However, the production of purely abstract theory has not been the motivating factor behind this work. The author believes that analytical philosophy can have very practical consequences and a second objective of the thesis is to give a demonstration of this fact.

A third objective is to present the theoretical and practical solutions in a way that can be easily understood by people working in the area of information system design. The main agenda will be to prove the point, a second agenda is to persuade professionals to use the ideas and adopt the proposed method in their work. It is for this reason that the arguments will center around a discussion of SSM. Unlike other methodologies used in information system design, SSM comes close to surfacing the problem of real world mapping. Also the core components of SSM can be developed to provide the logical requirements needed for a solution of the real world mapping problem.

In the next seven sections, sections 1.1.2 - 1.1.8, it will be explained how the thesis relates to the work that has already been undertaken in SSM and in information system design. It is contended that SSM is best pictured as a branching tree-like structure rather than something that occupies a pigeon hole. Given this the direction of the thesis can be plotted against the other branches of SSM. Problems that are now being encountered with traditional information system design methodologies suggest that it is worthwhile to look at radically different possibilities. SSM is radically different. However, neither Checkland nor Wilson, who were mainly responsible for the development of SSM, have capitalized on this difference in the context of information system design. This is not surprising as traditional information systems solutions are not capable of representing the unusual status of the SSM models. The full representation of the SSM models requires a new solution, this can be found by designing a system of the knowledge based type.

1.1.2 The nature of SSM

Any brief description of SSM will inevitably be both inadequate and contentious. An account of SSM must contend with a variety of authors, a variety of problems and a variety of results. Checkland and Wilson will be taken as the authorities on SSM in this thesis and it is their models that will be analyzed. However, it needs to be said that the applications emphasized by some authors are quite different from those emphasized by others.

At one extreme we have the work of Davis (Davis & Ledington, 1991) in which SSM is used as a kind of organizational psychotherapy with the methodology providing a structure for an on-going and endless debate. At the other extreme we have Wilson (1984, 1990) using SSM as a front end to SSADM with the SSM models providing the skeleton for detailed information system design. The use of techniques is similarly varied. In Avison & Wood-Harper (1990) and Patching (1990) heavy use is made of complex rich pictures comprising numerous icons. In Wilson rich pictures are not always used and when they are they tend to be frugal and non-essential. Checkland and Scholes' five constitutive rules for SSM (1990, pp. 284 - 290) are broad enough to cover many styles of use.

Nevertheless there is a feature which is common, and we can claim, fundamental to all uses of the methodology. This is the construction of a consensus conceptual model through an iterative debate driven by members of the client organization. A rich picture of SSM is given in figure 1. SSM is seen as a tree structure in which various types of problem form the roots, the conceptual model building process forms the trunk and various developments of this form branches some of which further divide before leading to different solutions. In the figure the solutions reached by Checkland and by Wilson are indicated as such, the other end points are solutions that might be obtained using the methods advocated in this thesis.

The tree analogy is a reaction against attempts to pigeon hole SSM as something contained in a particular area. Unlike cybernetics, SSM is a methodology not an area of study. SSM is not concerned with the study of systems but with development of system thinking which is part of the study of other things. SSM is similar to mathematics in that it is a way of thought that applies to many fields of study. It is dissimilar to chemistry which is an area of study that can employ many ways of thought e.g. mathematics, logic, systems.

Checkland and Scholes (1990, p. 284) have emphasized that SSM is a methodology rather than a method. The terms "SSM", "the methodology", "the use of SSM" and the "SSM method", will be used frequently in this thesis. These are not all interchangeable. SSM as a methodology is of a higher logical order to the use of SSM and the SSM method. The method exists in the wider context of the methodology. The methodology is *about* methods.

1.1.3 The direction of the thesis

It is hoped that the tree structure can be helpful to the reader by providing a rough guide to the direction that this thesis will take. The tree structure is just a guide line and should not be taken too literally. It is intended as a guide to this thesis and nothing more. It has a quite different status to the later figures which will be the subject of rigorous logical analysis.

Figure 1 shows results, at the end of the branches, and the models used to achieve them. The broken lines link the models and methods which have been devised during the doctoral research. Solid lines indicate traditional SSM methods and show how the thesis fits in with Checkland and Wilson's work. The branches to the left of the trunk show the sort of results obtained by Checkland. These results can be summarized as changes of thinking on the part of the people in the client organization. Outcomes of this type will only be tangentially related to this thesis. The main concern will be with those types of SSM project where the development of a rigorous logical model might lead to a useful outcome. Checkland's work will mainly be relevant to the initial model building.

The thesis will be more closely related to the work of Wilson where a detailed information system design is a possible outcome. It will be contended that an information system design needs to be robust and to have logical rigor. The author has considerable misgivings about Wilson's work both in terms of the rigor of his methods and the validity of the solutions. It is the latter that is of greatest concern.

The thesis will explore the consequences of an analysis of the basic SSM process from the perspective of logic, the theory of meaning and the theory of knowledge. Greatest emphasis will be given to the design of an information system of the knowledge based type as a possible outcome of an SSM project. This emphasis on knowledge based systems has not been the result of a plan. It has resulted from the fact that the research indicates that unless "information" is very narrowly defined, database approaches and structured methods have serious logical shortcomings as information system design methods. A method for designing a relational database *is* propounded but it seems that much of the content of a fully developed conceptual model must be lost in this type of solution.

1.1.4 Traditional ISD methodologies

The traditional methodologies, such as SSADM and Information Engineering, provide tools and techniques for professional analysts and designers. These professionals are wholly responsible for the resulting system. These methodologies take as their stating point either unambiguous specifications or empirical observation. It is upon this basis that the rigorous tools of analysis and design are applied.

Traditional methodologies are open to criticism on a number of grounds. The most simple is that in the environment of the 1990s they are failing to produce workable systems. There is ample empirical evidence for this. Given that this thesis is not an empirical work (see section 1.2) it is perhaps permissible, for the first and last time, to quote secondary sources. Jayaratna (1990) cites figures from the US General Accounting Office on the effectiveness of Federal software products; these indicate that less than 2% of software products were used as delivered, 3% used after change, 19% abandoned or reworked, 29% paid for but never delivered and 47% delivered but never used. Lyytinen (1988, p. 61) in a survey of empirical literature concerning information systems failures concluded that "Many reports show that somewhere between one-third to half of all systems fail, and some researchers have reported even higher failure rates".

The reason for these would also appear to be simple. SSADM and Information Engineering were developed as methods of computerization. Computerization can be defined as the process whereby a model is converted into a functioning computer system. In the case of SSADM the model has traditionally been an existing bureaucracy and the process begins with a manual systems flowchart. Information engineering is more subtle and uses the functional decomposition of existing processes as well as elements from the bureaucracy such as forms. Both however, are concerned with the representation of existing processes as their starting point.

The traditional methodologies have become victims of their own success. In developed countries most bureaucracies have now been computerized and therefore the demand for information system design is in green field situations. The availability of a host of packaged programs makes design unnecessary for standard green field requirements such as order processing and stock control. If the green field requirement is quite new then there will be no existing process for the traditional methodologies to use as a starting point.

These practical problems are rooted in serious theoretical difficulties which will be explained in a later part of the thesis. For the present it is enough to note that while various attempts are being made to bring these methodologies into line with current demand, their shortcomings are sufficient to justify serious consideration of other approaches.

1.1.5 Special status of SSM

A casual comparison of the use of SSM for information system design with the standard and traditional information system design methodologies reveals that SSM is startlingly different. An obvious difference is that traditional methodologies are analyst driven while the core of SSM is a stake-holder (client, actors and owners in, and of, the problem) driven modelling process.

Another major difference is that whereas traditional methodologies are based on some representation of the real world, the models in SSM are purely notional. This is quite clear in Checkland's writing. When Checkland talks about "holons" it is evident that he does not even regard systems as part of the real world.

> Choosing to think about the world as if it were a system can be helpful. But it is very different stance from arguing that the world *is* a system, a position that pretends to knowledge that no human being can have. (Checkland & Scholes, p. 22).

> The negative aspect is that the increasing complexity of the models might lead to our slipping into thinking in terms of models *of* part of the real world, rather than models *relevant to* debate about change in the real world. (Checkland & Scholes, p. 41).

Wilson also give a notional account of the SSM models:

> It cannot be emphasized too strongly that what the analyst is doing, in developing a HAS [Human Activity System conceptual] model, is *not* trying to describe *what exists* but *is* modeling a *view of what exists*. (Wilson 1984).

The term "real world" is apt to cause some debate as it may be thought that it is not clear what the real world is. The terms "external world" and "physical world" might also be used but there is also debate about what these are, and the three terms need not be regarded as coextensive. To pursue these questions would take the thesis far from its course which is not to determine the nature of reality. The author of this thesis takes the term "real world" to denote the world that is not illusory and which is not *made up*. This is sufficiently close to Checkland and Wilson for the purposes of the thesis which will be largely concerned with conceptual models which are *made up* and the physical world which is not *made up*.

Checkland & Scholes also think that as the models are not descriptions of part of the real world they are not valid either.

> Since the model does not purport to be a description *of* part of the real world, merely a holon relevant to debating perceptions of the real world, adequacy or validity cannot be checked against the real world. Such models are not, in fact, "valid" or "invalid", only technically defensible or indefensible. (Checkland & Scholes, p. 41, original italics).

The arrows in conceptual models are described as indicating "contingent upon" or "logically dependent upon". They are not intended to represent time dependencies as many people think. Checkland has specifically ruled this out in an unpublished internal discussion paper where he compared conceptual models with PERT networks which do indicate time dependencies.

> In such entities all arrows show logical dependencies; many of them also happen to indicate time dependencies. But it is the logic, not the time, which is fundamental. (Checkland, 1984).

It was remarks such as these, which indicate that the models have a unusual logical status, that prompted the research described in this thesis. It can be doubted that many SSM practitioners take Checkland literally here. However, if the word "valid" above is taken to mean factually true as opposed to logically true then we can take Checkland at his word and account for conceptual models in terms of modal logic. However, although modal logic can be used to explain what Checkland says about the models it will not explain what Wilson does with them.

1.1.6 Checkland on information systems

Checkland & Scholes claim that an information system:

> ...will always have to include the attribution of meaning which is a uniquely human act. An information system, in the exact sense of the phrase, will consist of both data manipulation, which machines can do, and the transformation of data into information by the attribution of meaning. (Checkland & Scholes, p. 55).

Checkland uses the formula "data plus meaning equals information". He does not seem to be very impressed by the data processing systems produced by traditional methodologies. Nevertheless, he would be unlikely to contend that the traditional methodologies are an efficient means of computerization. His point is that a data manipulation system is not a information system. His work in this area, is therefore, concerned with the design of information systems of which a computer system may, or may not be, a part.

Checkland's work is quite consistent in this area. He produces human activity models that are relevant to information problem. He also produces models of systems to build information systems (see Checkland, 1989). What he does not do is use SSM to produce a design for a computer system that can work within the information system, nor does he think this has been accomplished:

> Much work is currently underway in this area. In particular SSM could enrich those poverty-stricken stages of systems analysis and design methodologies in which information requirements analysis is assumed to be straight-forward, or organizations are naively documented as a set of unproblematical entities and functions... (Checkland & Scholes, p. 55).

Checkland makes two assumptions here that will be challenged in this thesis. One is that attribution of meaning is uniquely human. The second, which in Checkland's terminology will follow from the first, is that computer systems can only manipulate data. Checkland's system of thought here could be understood as indicating that "meaning attribution" will include real world mapping. He seems to be concerned about reference here. He could be interpreted as saying that item of data does not have reference unless a reference is ascribed to it by a human being.

As reference is necessary for real world mapping, it follows that humans will map the data on to the real world and only humans can map the data onto the real world.

There are two problems with this. One is that it relegates computers to an unnecessarily limited role and ignores work on artificial intelligence. The second is that the people who interface with the computer do not always critically map the data on to the real world. They often, as it were, take the computer's word for it. This produces a strange situation where the computer indicates that something does not exist, the operators believe the computer, but it is plain for all to see that it does exist. This problem is not unique to information systems involving computers nor is it new. Fifty years ago Franz Kafka's novels explored the paradoxical situation where a bureaucracy (read: "information system") and the behavior of the people involved with it did not correspond to perceivable events.

The thesis will attempt to show that computerized information systems are capable of taking on more of the real world mapping task than Checkland and many others might think. This opens up the possibility of computers being capable of meaning attribution. Of course, if human activity is built into the definition of "meaning" this will not be possible. However, if a decision on whether computers can attribute meaning is left open to something like the Turing test (Turing, 1950) then the system that is constructed in chapter 6 might be said to exhibit a rudimentary form of meaning attribution.

1.1.7 Wilson on information systems

There is a conflict between Checkland and Wilson over SSM and information system design. While Checkland thinks that the link between SSM and detailed computer system design has not been accomplished, Wilson offers a step by step method to accomplish it.

In the earlier sections it was argued that the present shortcomings of traditional methodologies warranted consideration of a radically different approach and that SSM was radically different. The first thing to note about Wilson's work is that his solutions are of the same type as those of traditional SSADM. It is, therefore, implicit in Wilson work that there is nothing wrong with traditional solutions. For Wilson the problem must be the way in which the solution is obtained not the type of solution itself. This thesis will argue that these types of solution cannot solve the real world mapping problem and are, therefore, inadequate. However, it will also be contended that the SSM conceptual models, if suitably developed, are capable of producing a computer system design that can deal with the real world mapping problem. Wilson's method can be criticized for not taking advantage of this.

Real world mapping is a double theoretical error in Wilson's work. Not only does he fail to produce a computer system that can deal with it, he also fails to account for it in the internal working of his method. He starts with a notional model and proceeds to produce a computer system that will monitor and control stock in a warehouse (Wilson, 1984, pp. 195 - 208). It is obvious that our notions do not always correspond to real world facts, if they did we would never revise our opinions. Yet Wilson presents the consensus primary task model as incorrigible. It is something which through a series of steps (construction of information categories, Maltese cross and data flow diagrams) will inevitably produce a computer system that represents physical objects.

It is easy to see how this muddies the water with regard to the logical status of the conceptual models. Given that we end up with computer code that represents physical objects, and given that the code is derived from a conceptual model, then it would seem that the conceptual model must also represent physical objects. But as Wilson himself points out the conceptual models "is *not* trying to describe *what exists* but *is* modeling a *view of what exists*." To solve this dilemma requires a solution to the real world mapping problem, Wilson does not provide a solution nor does he even recognize the problem.

It might be thought that a solution can be found on the fringe of SSM. The constructivists argue that everything is a mental construct. As everything is an idea there is no separate real world to map ideas against and thus no real world mapping problem. (Stephens & Wood (1991) argue for a form of constructivism and link this to SSM. Their position is that communication, rather than an individual's ideas, is ontologically formative. Thus, for them, communication creates things in the real world and there is, therefore, no need to map communications on to the real world.)

Powerful objections to this position will be given in a later section, for the present it is enough to point out that even if correct the constructivists would not solve Wilson's problem. We would still need to distinguish between ideas that are voluntary and those that are not. We would have to account for the fact that some voluntary ideas such as a plan for a nice day at the sea side to not always map onto to involuntary ideas such as the weather.

The special logical status of the SSM conceptual models is inevitably lost in Wilson's method because of the limited logical power of the solution. Despite this Wilson's method may still be a useful practical guide and a considerable improvement on traditional methods. Although the logic does not carry through into the solution the building of the conceptual model may still have a causal effect on the design. It can, causally, make the designers of traditional solutions more sensitive to the context in which the solution is intended to operate.

1.1.8 SSM for knowledge based system design

The thesis will describe a method whereby an SSM conceptual model can be developed to the point where a computer program can be derived. The program will be the basis of a computerized information system that will have learning capability. This informations system will be best described as a knowledge based system. The term "knowledge based" is appropriate because the system will be set up to include empirical knowledge and have the capability to acquire more empirical knowledge. It can, therefore, be distinguished from those expert systems which comprised only definitions and in which the rules are permanently fixed. It will also be open to data from the real world and so can be distinguished from closed world artificial intelligence systems such as chess players.

At one level the thesis will present an outline and a proposal for a practical method of knowledge based system design. This, it is hoped, will be interesting in itself. It presents a powerful extension of Soft Systems Methodology. It is a development which will enable SSM to be used in the production of intelligent computer systems, and therefore gives it the capacity to produce solutions which have hitherto been beyond the scope of the Methodology. The method will enable the computer system to dynamically incorporate the language of the users and, therefore, to perform operations to go beyond mere data manipulation. Such a computer system will be capable of being a true information system in Checkland's sense of the word.

At a second level it surfaces a number of theoretical problems in SSM and in traditional information system design. These problems are identified and explained. Once explained a practical solution, which is theoretically sound, can be offered. This might be instructive to design practitioners whose practical problems are sometimes the result of theoretical difficulties that are not even recognized. Once theoretical gaps in Wilson's method and the Multiview method (Avison & Wood-Harper, 1990) have been identified the problem situation in which they can be successfully applied is more easily recognized.

At a third level it presents one solution, there may be others, to the real world mapping problem. The method outlined at the first level can be taken as an example which illustrates how the real world mapping problem can be solved. The benefits of a computer system that can achieve this should be fairly obvious. It represents a general increase in the power of the computer system. It will stand to reduce error in the interface between the computer system and the physical world. It will allow the computer system to respond to changes. It will enable designers to cope with green field situations where information requirements are novel.

1.1.9 Plan of the Thesis

These three levels of inquiry will largely be interconnected and, therefore, they will not be dealt with sequentially in the substantive sections. The introductory chapter continues with a description of the methodological issues that bear upon the thesis, a comment on the literature and an explanation of the limitations of the study. The main arguments are put forward in chapters 2 through 6. Each of these main chapters deals with a theoretical problem, finds a solution that is theoretically sound and suggests a practical method for the implementation of that solution. The main chapters are self contained arguments that are of interest in themselves. Each section has been published in one form or another. Each main chapter produces conclusions that lead into the following main chapter. Many sections also produce conclusions and suggested avenues for research which are not followed in the remainder of the thesis. Returning to the tree analogy in figure 1, these are branches which will not be explored here. They do, however, give extra weight to the contention that the line of inquiry is useful. The final four chapters give on outline of potential scientific applications for the method, a description of a project using the method, a account of some philosophical arguments that bear on the subject, and conclusions.

The thesis is organized into four parts. Part I comprises the present introductory chapter. Part II comprises chapters 2 - 7, this contains the main argument of the thesis presented as a continuous narrative and concluding with an account of an action research project. The main objectives of the thesis are accomplished by the end of chapter 7. Part III contains ancillary arguments which support some of the arguments made in earlier sections and explore some arguments in greater depth. Many of the arguments in Part III are inconclusive, indeed some of the philosophical argument have been going on for hundreds of years. Part III is speculative, when compared to Part II, and seeks to link the earlier work to a wider debate. Part IV consists of the conclusions and directions for further research, this is of considerable importance because the wide ranging research program indicated in the thesis may well be of greater importance than the work accomplished in it.

Chapter 2 shows how SSM conceptual models can be expressed in the propositional calculus. It explains that the logic of SSM models is not capable of representing cause and effect which is tacitly required in Wilson's method. It is shown how the models can be enhanced, by the addition of two logical connectives, in such a way that they become capable of representing causation. The benefits of doing this are numerous: it enables SSM to be used for cause & effect diagramming; it leads to a logical, rather than a mathematical, account of efficiency; it shows how SSM can be used for automating monitor and control activities; it enables the models to be converted into analytic data flow diagrams thereby effecting a link to structured solutions in information system design. However, problems remain in the area of information system design because propositional logic, like structured solutions, is incapable of representing the notional status of the SSM models.

Chapter 3 explains the notional status of SSM models in terms of Wittgenstein's language game theory. A logical foundation for information system design requires a theory of meaning. The section considers Ideational theories which attach meaning to the ideas in the private world of a conscious subject. These are contrasted with Wittgenstein's theory which held that language and meaning were primarily public and that a private, purely subjective, language was impossible. The iterative debate among stake-holders that takes place in the practice of Soft Systems Methodology (SSM) can be understood as a Wittgensteinian language game in which meaning is created not just discovered. If conceptual models are understood as stipulative definitions then they can be expressed as logically true in modal logic. With this the notional status of the model can be expressed rigorously in a formal system.

Chapter 4 introduces predicate logic in order to enhance the limited logic of chapter 2 but also to answer criticism of SSM put forward by Probert. A distinction is made between universal statement and particular statements. Three types of universal are identified: inductive hypotheses, value statements and definitions. A second line of argument is presented to show that SSM models are sets of definition. Multiview models by contrast are sets of inductive hypotheses. Therefore, in modal logic SSM models have the status of being logically true while Multiview models have the status of being factually true or false. As inductive hypotheses and definitions are inexorably tied together neither type of model is adequate in itself.

Chapter 5 builds a model that combines the logically true elements with factually true elements. Thus merging Checkland and Wilson style models with Multiview style models. The composite model demonstrate how real world mapping of a notional model can be achieved.

Chapter 6 introduces the six stage method for knowledge based system design which constitutes the main deliverable of the thesis. The method begins with system analysis and proceeds through stages of language creation, knowledge elicitation, knowledge representation, codification and terminates with an in-built mechanism for verification. Each stage, with the exception of the last, comprises the development of a model. The model building process begins with an SSM conceptual model which develops into a logico-linguistic model, then an empirical model, a modal predicate logic and, finally, a program written in Prolog.

A project for problem structuring in a public relations company is described in chapter 7. Logico-linguistic modelling was used in the project up to the knowledge representation stage. It was found that the client did not have sufficient knowledge of the problem area to build a worthwhile knowledge based system. Instead the model was used to put together plans for empirical research projects that would increase the client's knowledge. As a demonstration the chapter continues by describing how a relational database and a knowledge based system could have been obtained from the model.

The example given in chapter 6 is a rather linguistic one. It was mainly concerned with how a given situation should be described. Chapter 8 gives an example that shows more clearly how the method can aid in scientific discoveries. The foundations of formal modal logics are discussed and informal system of rules for modal operators is described.

Chapter 9 discusses a number of philosophical issues that have arisen during the proceeding chapter. A comparison of logico-linguistic modelling methods with other forms of knowledge representation is also made.

1.2 Methodology

1.2.1 Philosophical analysis

Business studies is not a discipline but an area of study. A thesis in this area must either develop its own methodology, a bold undertaking which will not be attempted here, or take a methodological position from another discipline. It is normal in a business studies degree to take the methodology from the social sciences and in business studies concerned with information systems sociology is probably the favorite. This thesis will break with this tradition and take modern analytical philosophy as the parent discipline. The methodology will, therefore, be one of analysis from the perspectives of philosophical logic, the philosophy of language and the theory of knowledge. This can be justified in two ways. The first line of argument is to show that sociology is not an inevitable choice, the second is to show that philosophy is appropriate.

Choosing to take the methodology for the study of business information systems from sociology is fraught with difficulty because sociology has, and always has had, serious methodological problems of its own. It is difficult to think of another discipline where the practitioners are in such wide disagreement over what their subject is about. The methodological stances of Weberians, Durkhiemians and Marxists are fundamentally apposed and incompatible. If sociology is taken as the parent discipline then we will have Weberian business information studies, Durkhiemian business information studies and Marxist business information studies. Anyone using one of these methodologies would have to justify the choice and this justification would require an appeal to the philosophy of science. This is not to say that using a methodology drawn from sociology would be misconceived, it is to say that drawing a methodology from sociology is not as simplistic as many might believe.

Advocates of Human Centered Design argue that the fulfillment of human needs must be taken as the main objective of computerized information systems. This would indicate that sociological findings are relevant to system design but it does not indicate that sociology should provide the methodology for system design.

The choice of social science methodologies as the foundation of information system design is by no means universal. Some argue that computer systems are essentially about communication and language. In some areas of cognitive science and artificial intelligence linguistics provides the major methodological input. Cybernetics takes most of its methodology from biology.

An argument can be made for the appropriateness of philosophy by considering the nature of current work in information systems. Judging by the papers that appear in British information system journals the subject is becoming increasingly philosophical. However, few of the writers on the philosophical aspects of information system design have any formal training in philosophy.

It is pertinent to note that the line of inquiry of which this thesis is a product began with a MSc dissertation about SSM for the Department of Systems and Information Management at Lancaster University. The author was encouraged in this work by the late Ronald Anderton, then Head of Department, who thought that the work would be interesting because nobody trained in philosophy had ever undertaken an analysis of SSM. Considering the range of main stream philosophical ideas discussed in Checkland's "Systems thinking, systems practice"(1981), this was somewhat surprising. It would appear that the use of analytical philosophy as a methodology in this area of study is not only appropriate but long overdue.

It should be made absolutely clear at this point that the methodology will be that of Anglo-American analytical philosophy. This has very little common ground with Western philosophers of the Continental School. In the British information system journals most discussions of philosophy refer to philosophers of the Continental School. This can be explained by the fact that Continental philosophers are popular in the social sciences and that writers on information systems have become familiar with these philosophers though the social science literature. This thesis is written in the belief that analytical philosophy has far greater relevance to information systems than

Continental philosophy. However, there will be no attempt to argue this point. The thesis will attempt to show rather than say. That is, the thesis will use the tools and techniques of analytical philosophy and produce results, such as a program in Prolog, that could not have been produced using continental methods.

1.2.2 Systems as its own methodology

It could be argued, has Checkland has done (1981), that Systems Studies is, itself, an emerging discipline. If this is correct then Systems could be taken as the methodology for the thesis. However, as the thesis will be concerned with systems methodologies this would be circular. One would be using systems ideas as a methodology to critically examine systems ideas.

Unfortunately the use of philosophy as a critical tool to does not entirely avoid this problem. If Checkland is correct "Systems" is a school of philosophy - a school committed to certain philosophical positions such as the existence of emergent properties. Given this the circularity returns. We will be using philosophy as a methodology to critically examine philosophy. As philosophy is a study of the highest logical order (it takes nothing as a given while all other disciplines must take as given a theory of knowledge and a theory of truth) there is no easy way to avoid the problem and any attempt at a solution would take us well beyond the scope of the present work.

What happens in practice is that one area of philosophy is used to critically examine another. Thus the philosophy of language might be used to examine a theory of knowledge, and the theory of knowledge might be used to examine a theory of meaning. This is the solution that will be adopted here.

1.2.3 Limits of current systems thinking

Checkland (1981) and Churchman (1971) have produced works which are relevant to information system design and have a high philosophical content. Both are concerned mainly with epistemology - the theory of knowledge. Both attempt to give an account of knowledge and having done so proceed to give an account of the nature of the world. When they discuss the philosophy of science, ontology or metaphysics their arguments are drawn from their epistemology. However, neither offer anything particularly new in this area.

Churchman's book is treated with respect and something like reverence in systems circles, but for a philosopher it is a rather pedestrian history of epistemology. The book has not occasioned much interest in philosophical circles. While it provides a useful function in drawing attention to the importance of epistemology for an information systems audience, it does not solve, or even attempt to solve, the major problems in information system design. Although Checkland is not a professional philosopher his work is, as this thesis will seek to show, of considerable philosophical interest. Perhaps more than that of Churchman who is a professional philosopher. However, the significance of Checkland's method lies not in epistemology but in the philosophy of language. Although epistemology will inevitably be bound up in the this thesis the main tools of analysis will be the philosophy of language, the theory of meaning, philosophical logic and modal logic. The lines of demarcation now need to be drawn.

The traditional analysis of knowledge is that it is true, justified belief. Some attribute this analysis to Plato (Gettier, 1967) and although some philosophers think that this analysis is in need of considerable qualification, it will be sufficient to explain the relationships between philosophical logic, epistemology and the philosophy of language. The following is a very simplistic explanation.

Theories of truth are a concern of philosophical logic. Historically there have been two main theories. The correspondence theory of truth states that a proposition is true if it corresponds to a fact. The coherence theory of truth states that a proposition is true if it coheres with other propositions that are true.

Justification is a concern of epistemology. Historically there have been two main theories. Rationalists, such as Descartes and Spinoza, state that a beliefs are justified by the understanding of the logical connection between them, rationalists hold that knowledge must be *a priori*. Empiricist hold that all knowledge comes from experience and that logical connections are just tautologies.

Belief is a concern of the philosophy of mind and the philosophy of language. Built into the notion of belief is assent to a proposition and therefore an understanding of what it means. The philosophy of language is largely a twentieth century development and there are no clear cut historical positions. However, there is a divide about how to deal with meaning. Semantic theories seek to explain the meaning of a proposition in terms of its truth conditions, for them the theory of meaning loops back to philosophical logic. Other theories, notably that of the later Wittgenstein, seek to explain truth in terms of meaning.

A word of warning about terminology is appropriate here. A person familiar with Checkland's work but not familiar with philosophy will be confused by the use of the term "epistemology" here just as a philosopher is apt to be confused by Checkland's use of the term. Checkland uses the term "an epistemology" to denote a framework in which something can be learned rather than as a term to denote high level theory about what is and is not a valid account of knowledge.

1.2.4 Misconceptions about logic

Logic is an integral part of philosophy and will form a major part of the arguments that follow. The remainder of this methodology section will try to explain what it is (in information systems writings one finds the most peculiar ideas about it). We can start with some general definitions: "Science of reasoning, proof, thinking or inference" Concise Oxford Dictionary; "the study of Inference" Fontana Dictionary of Modern Thought; "In its broadest sense logic is the study of the structure and principles of reasoning or of sound argument." Dictionary of Philosophy (Flew, 1979).

Logic is about inference or reasoning. The adjective "logical" refers either to the science or to an inference that is valid in logic. There is, however, a slight equivocation in the use of the word "logical" even in the rigorous world of philosophy. Normally we will say that an argument is logical if it is deductively valid, but logic textbooks often contain sections on induction or the logic of commands. However, David Hume's notorious problem of induction is based on the contention that induction is not deductively valid. Nor is the logic of commands deductively valid (see Probert, 1991, 1992, for a recent discussion). In one of the later substantive section when modal logic is introduced the term "logically true" will be used to describe definitions, but definitions are not the result of valid deductions. To determine the exact range of the term "logic" requires recourse to theories of truth and theories of meaning and these theories are all contentious.

In spite of this there is a large amount of agreement about logic in academic philosophy and mathematics. It is generally agreed that the term "logical" applies to propositions, statements or arguments made up of propositions or statements. The term "logical" never refers to real world events themselves, it can however, refer to statements about real world events. This is in contrast to ordinary usage where the term "logical" is equated with "reasonable" or "rational" and as such can be used to describe real world behaviour directly.

This unfortunate difference between technical and popular usage can leads some people into serious error. It leads some people to think that rules of logic are intended to be laws of nature. It leads some people to confuse logical connectives with causal relations. This can be illustrated with a syllogism:

All men at the draw won a lottery prize
Socrates was a man at the draw
Therefore:
Socrates won a lottery prize

The conclusion follows from the premises by the rules of *Modus Ponens*. The conclusion will be true if the premises are true. More importantly it will be true that the conclusion follows from the premises even if the premises are false. Deductive logic alone cannot generate facts about the physical world. Some people take the example to be a causal account, they think that it is trying to say that being a man at the draw caused Socrates to win. But this is not so, it could be just coincidence that all the men won; this would not make any difference to the logic.

A second misunderstanding about logic is more complex. It has been said by commentators on earlier versions of this work that logic is only about formal systems. This is not true. Although logic can be used to describe formal system, such as mathematics, it can also be used to describe a casual conversation in a pub.

Yet another view, which is not endorsed in this thesis, is that logic is only a formal system and is quite distinct from meaning. On this account meaning is a matter of semantics, logic is a matter of syntactics, syntactics is only concerned with formal systems and, therefore, logic does not play a part in determining meaning.

The system that will be used here is as follows: Meaning comprises sense and reference. Reference includes the connection between terms and things in the real world. Reference is determined by semantics. Sense is about the connection between terms and other terms. The connection between terms and other terms is governed by the rules of a Wittgensteinian language game. Syntactics is concerned with rules. Sense can be determined by reference i.e. semantically, or by rules i.e. syntactically. Any language formal or otherwise is bound by rules. Therefore, syntactics can play a part in determining meaning in any language.

The thesis does not accept the idea that the semantics of a set of terms must be established before a syntactic system can be devised. That reference must be established before sense. The argument that will be put forward here is that the syntactics of a set of terms must be established before the semantics. That sense must be established before reference.

A third misconception about logic comes about through deviant usage in computer science. Data flow diagrams are described as logical. They are however, nothing more than an abstraction from manual systems flow charts. They are like the standard London Underground map which is an abstraction of a scale map of the Underground. There is nothing logical about them, though they could be a stage on the route to logic. The term "logical" in this context seems to mean nothing more than "not physical". This deviant usage like many others in computer science, for example "relation" in relational data base design, is very unfortunate. This is because some diagrams in artificial intelligence, such as Sowa's conceptual graphs, are intended to be logical in the true sense of the word. It would not be far from the truth to say that they are pictorial versions of the predicate calculus (they are not always exactly coextensive, but this is a complex point that will be dealt with later).

Discussions of logical models represented by diagrams will be a major part of the substantive sections below. It must be understood that "logical" means a lot more than "not physical". The fact that something is not physical does not mean that it is logical.

1.3 Limitations of the Study

1.3.1 Philosophical points of controversy

Although the substantive sections will contain a good deal of high level theory and will seek to establish a number of theoretical points, the objectives here are not solely confined to theory. There will be an attempt to show how the theory can lead to a practical method for knowledge based system design. This means that a lot of ground will have to be covered in a short space. In philosophy there are few if any facts or theories that are not contentious. It will not be possible to discuss all the theories that bear upon knowledge based system design. Philosophical issues will be raised only when they have immediate bearing upon the development of the argument. There will be an attempt to base the arguments on respectable philosophical positions.

That is positions that although they do not necessarily command general assent, are not generally considered to have been discredited.

These days argumentative PhD theses tend to be written defensively. The authors take a minor point and try to establish it incontrovertibly. This will not be the tactic here. The arguments themselves will, hopefully, been seen to be valid however the premises upon which they rest will inevitably be contentious and open to dispute.

1.3.2 Formal problems in information systems

Logic and theories of meaning have application in information systems design in connection with their formal properties such as completeness, decidability and validation. These are implementation problems and will not be the main area of concern in the thesis. The main area of analysis will be in systems analysis and knowledge elicitation. The logic employed does bear upon the formal properties involved in implementation and the thesis will show how this logic carries through into a program. However, a full discussion of all the formal issues that bear upon computerized information system design will be beyond the scope of the thesis.

1.3.3 Results

The practical guidelines for the proposed method for knowledge based system design go as far as the theory will usefully take it. Stake-holder constructed Logico-linguistic conceptual models are an essential tool in the proposed method. These are far more complex than the SSM conceptual models. The theory indicates that they are necessary but provides no indication of how their construction might be managed. The assumption is that they can be constructed in more or less the same circumstances as those in which traditional SSM models are constructed. However, the way to find out would be by empirical research. Empirical research in this area would be best carried out in the context of action research, as has been argued by SSM advocates. Such research projects are very time consuming and beyond the scope of the thesis. This work would be a development and implementation stage that would follow the theoretical work undertaken here.

The case study at the end of the thesis does give some empirical support to the contention that the method is a practical possibility. However, this empirical support is minimal. It only shows that Logico-linguistic models can be understood and constructed by some businessmen that have no training in logic, philosophy or information systems. The main reason for the case study is to provide a more natural example of how the method might work. Conceptual models and Logico-linguistic models are intended to be constructed by people in the client organization. The examples in the substantive sections are made up. This is rather a strain. SSM conceptual models are not models of the real world. A conceptual model made up as an example is not a real conceptual model but a hypothetical example of what a group of stakeholders might have come up with if they had really been there to build the model. It is thus, two stages removed from reality.

It was found in presentations that these hypothetical examples often occasioned dispute. In one case a hypothetical conceptual model of an order processing system was vehemently attacked because "order processing systems are not like that". This objection was irrelevant as a conceptual model of an order processing system is not intended to represent any existing order processing system. A more pertinent objection would have been "a group of people involved in order processing would never build a conceptual model like that". This might have been true, but does not matter as the point of the hypothetical model was not to show what models stake-holders would actually build but to illustrate the logic involved in the building process. Logical points are best illustrated in very simple models, but very simple models are rarely credible examples of what stake-holders might build.

The thesis puts forward a method and a theoretical justification of the method. It is, therefore, distinctive in the field. Wilson and Avison & Wood-Harper put forward methods but little in the way of justification of the methods in terms of the theories knowledge and meaning. Checkland in the context information systems puts forward a lot of theory but little on method. Although Checkland's Systems Thinking, Systems Practice (1981) might provide a general justification for Wilson's line of approach, the techniques such as information categories and the Maltese cross have no substantial theory behind them. In the following substantive sections each move in the method is given a theoretical justification.

1.4 The literature

It is conventional in a thesis in Business Studies to include a literature review as part of the introductory sections. The reason for this is because Business Studies tends to follow the methodology of the sciences especially social science. The method is different in philosophy as it is concerned with developing arguments rather than evaluating evidence. In a philosophy essay relevant literature is not assigned to a separate section but is discussed throughout the essay as it becomes relevant to the developing argument. As this thesis will follow the methodology of philosophy there will not be a separate literature review.

There are also practical reasons for omitting a separate literature review. If the review was given a narrow domain then it would be largely limited to a review of the author's own work. The logical analysis of soft systems modelling is an unprecedented endeavour. The first paper on the subject was the author's *Causation and Soft Systems Models* (see declaration above). Since the publication of that paper there have been other authors on the subject, Probert (1991, 1993) and Merali (1993), and their work has been the result of the author first drawing attention to the subject.

If the review was given a wide domain and concerned itself with logic and information systems it would be so large that a bibliography alone would take up an entire PhD. Digital computers are logical machines and, therefore, any paper that discussed computers would be included in the domain. Even if the review were limited to papers that explicitly mention logic and information systems the domain would still be too large - there are some journals entirely devoted to this subject e.g. the Journal of Logic Programming.

The writings of a "philosophical" nature on the subject of SSM which appear in the information systems literature have little relevance to the rigorous logical analysis that will appear in the thesis. Of far greater significance is work in artificial intelligence which has been completely unconnected with SSM; some of this will be discussed in later sections.

PART II

2 CAUSATION

2.1 Introduction

During the last decade Soft Systems Methodology (SSM) has had considerable success as a general purpose problem solving methodology. The ability of SSM to address unstructured (soft) problems can be contrasted with traditional Operational Research which aims at solving structured (hard) problems. Although there is a contrast this does not amount to an incompatibility. During the process of SSM analysis a soft problem will often turn into a hard problem which can be solved by structured methods.

A key device in SSM is the development of a conceptual model. As was stressed in section 1.1.5 this type of model is not intended to represent what exists. The difficulty here is that having constructed a desirable conceptual model there is no guarantee that it will correspond to anything that *actually can* exist. In most uses of the methodology, especially Checkland's, this does not matter as the primary aim is a change of perspective on the part of those concerned rather than a change in a state of affairs in the physical world. However, Wilson and Avison & Wood-Harper use the models as the starting point for the design of information systems intended to support physical processes. In these cases a correspondence between the models and the physical world is required.

Any such correspondence is inhibited by the logical poverty of the models. The models contain only one type of logical connective and this is incapable, in any straight forward way, of fully representing causation. Without a comprehensive representation of causation there can be no guarantee of a correspondence between the models and the physical world. This chapter will show how the logical power of the models can be increased to the point where they are capable of representing causation and the physical world.

In section 2.2 an analysis of the logical status of SSM conceptual models is undertaken. It is suggested that a correspondence between the models and the physical world will hold if two conditions are met. Firstly, the terms or elements of the model must refer, directly or indirectly, to objects or events in the physical world. Secondly, the relations between the terms or elements in the models must have the same logical form as the relations that hold between the objects and events in the physical world. It is in respect of this last condition that SSM models are inadequate. The elements in the models are connected only by relations of necessity. Relations of sufficiency are also required in order to match the causal sequences in the physical world.

Section 2.3 shows that sufficiency can be introduced into the models without difficulty. With this additional relation a logico-linguistic model is produced which is exhaustive and capable of representing, within the limitations of the propositional calculus, any conceivable state of affairs.

Section 2.4.1 considers applications resulting from the fact that a logico-linguistic model functions as a conceptual cause and effect diagram. The ability to represent all logical possibilities in a cause and effect sequence gives the model greater scope than the empirical models that tend to be used in quality control. Section 2.4.2 discusses applications concerning efficiency. In SSM, a criterion for efficiency is one of three measures of performance that accompany every system, but it is not clear what this criterion is meant to operate on. It is suggested that efficiency can operate as the arbiter between two or more conditions that are sufficient for a desired effect. This gives efficiency a systemic and logical role that can be complimentary to quantitative accounts of efficiency in terms inputs and outputs.

Cause & effect diagramming and the logical account of efficiency are not taken any further in the thesis, the remainder will be concerned with information systems and knowledge acquisition. Section 2.4.3 describes how the use of propositional logic can pay immediate dividends in information system design.

2.2 Conceptual models in SSM

2.2.1 The outcome of using SSM

General descriptions of Soft Systems Methodology (SSM) are highly diverse. SSM has been characterized as a learning system (Checkland, 1985), part of a new paradigm for Operational Research (Rosenhead, 1989) and as a front-end for information system design (Curtis, 1989). However, such diversity is to be expected considering that its aim is to address any kind of unstructured "soft" problem in any organizational or social context.

SSM functions as a learning system because it facilitates a greater understanding of the problem situation on the part of those concerned. By bringing out the world views (Weltanschauungen) of the people involved in the problem situation, SSM can produce various types of result: the problem might simply disappear as the result of a consensus; a fairly unstructured solution might result, such as agreement to adopt a new role for the organization; a third possibility is that the problem becomes structured, in this case a soft problem resolves into an identifiable "hard" problem. It is this third type of result that will be the subject of most of the thesis.

2.2.2 The seven stage process

The classic SSM method is a seven stage process comprising: (1) entering the problem situation, (2) expressing the problem situation, (3) formulating root definitions of relevant systems, (4) building conceptual models of Human Activity Systems, (5) comparing the models with the real world, (6) defining changes that are desirable and feasible, and (7) taking action to improve the real world situation (Checkland, 1989; Checkland & Scholes, 1990; Wilson, 1984).

The dynamics of the method come from the fact that stages (2) through (4) are always an iterative process. The stake-holders engage in a debate guided by the analyst/facilitator. (In this thesis the term "stake-holders" is used only as a shorthand for the Client, Actors and Owner elements of Checkland's CATWOE). During this debate various root definitions (succinct statements of appropriate systems) and conceptual models are put forward, modified and developed until a desirable model is achieved by consensus. This model then forms the basis for real world changes.

2.2.3 Models and concepts

The name "conceptual model" is ambiguous. It could mean a model of a concept or it could mean a model that is conceptual. The notional status of the models would seem to imply that they intended to be models of concepts. However, it is worth reflecting on the difference here.

We can distinguish between *what models are* and *what models are models of*. With the exception of iconic models, such as a scale model of Winchester Cathedral, most models *are* concepts. But they are, mostly, intended to be *models of* real world states of affairs. The value of a model is usually directly proportional to how well it corresponds to a past, present, future, actual or potential state of affairs. A *model of* a concept is quite different because in order to be a good model it need not have this real world correspondence.

One of the features of modelling concepts is the ability to represent notions that have no easily defined physical equivalent. Rules, laws, values, and judgements can easily be represented. This, plus the ability to represent, compare and integrate various Weltanschauungen, gives models of concepts tremendous scope. This scope must always be greater than the scope of a model of a physical state of affairs for the simple reason that models of concepts are limited only by what is conceivable. As no model of a physical states of affairs can be inconceivable, every model of a physical state of affairs must be capable of being paired with a model of a concept.

This unlimited modelling scope allows Checkland to achieve solutions that could not have been identified using models of actual states of affairs. This is particularly true where the problem has been non-physical i.e. a problem about goals, gaining a consensus, values etc.

2.2.4 Concepts and the physical world

In subsequent chapters the idea of conceptual models being models of concepts will need to revised and refined, for the present it can be taken as a given. This prompts a consideration what sort of relationship can exist between a conceptual model and the physical world.

When a physical solution is required to resolve a problem situation, Checkland does not, in practice, take the models far beyond a general description. This is clear in the results of case studies. In Checkland and Scholes' *Soft Systems Methodology in Action* the outcome of the case studies are described as changes in thinking or perspective, changes of role for the organization as a whole, problem identification, or what has been learned about the organization. While these changes in thinking have lead on to real world changes such as detailed organizational restructuring and new information channels, the real world changes were not specified by the methodology.

By contrast, the case studies in Wilson's *Systems: Concepts, Methodologies and Applications* come up with specifications for new information processing procedures intended to support physical processes. This implies that there must be some relationship between the conceptual model and the information system; just as there must be some relationship between the information system and the physical process.

There is a *prima facie* dilemma here. What has been said so far is that a conceptual model need not be based on anything in the physical world. If this is so, then it would seem to follow that there can be no guarantee that a desirable conceptual model will ever correspond to anything in the physical world. The literature of SSM is not enlightening on this point. It is not clear how the transition from a model of a concept to a change in the physical world is effected. At stage (4) in the methodology we have a model of a *view of what exists*, but *a view of what exists* might bear no relation at all to what actually exists. In this case, the model can be no help in taking action to improve the real world situation as is required in stage (7).

The most simple solution to the dilemma is to take the conceptual models to be a number of inductive hypotheses connected together.

Given this, the models would be empirical and could be tested against events in the real world. However, this would mean that they are not models of concepts but models of putative physical states of affairs. As such they would not be significantly different from most other types of model.

A second solution to the dilemma is more difficult. This is to take the conceptual models as being logico-linguistic models. On this interpretation model building is a type of Wittgensteinian language game in which the stake-holders create an agreed language for describing the problem situation (this account will be detailed in chapter 3). The iterative process enables the sense (connotation, intension) of the various terms in the models to become fixed, thereby establishing a syntactical structure. In this way the models are analogous to formal systems such as arithmetic.

There are two requirements for a formal system to correspond to the physical world. The first is that its terms should have direct or indirect reference (denotation, extension) to objects, events or states or affairs in the physical world. The second is that the functional connectives should be capable of reflecting the behaviour of objects, the sequence of events or changes of states of affairs in the physical world. In arithmetic the terms are numbers, the functional connectives are addition, subtraction etc. In SSM models the terms are contained in the bubbles and the functional connectives are the arrows between the bubbles.

Exactly how reference is established has been the subject of ongoing debate among some of the world's most eminent philosophers and logicians for nearly a century. However, although reference is difficult in theory it tends to be unproblematic in practice. If people can agree about the sense of a word there is usually no problem about establishing whether it has reference or not. In the present case, sense is unproblematic because it is established by building the model.

This leaves the second requirement. The most general principles governing the behaviour of objects, the sequence of events and changes of states of affairs in the physical world are the laws of cause and effect. The remainder of this chapter will show firstly, that the connectives in SSM models do not reflect the laws of cause and effect, secondly, that this shortcoming can be easily avoided by a modification of the models, thirdly, that such modification could have interesting applications.

2.3 Propositional Logic

2.3.1 The logic of soft systems models

According to Checkland and Wilson the SSM modeling language consists of English verbs. These are formulated into elements which express commands. This has the advantage of being easily understood by the stake-holders in the client organization and this is essential as their participation is a fundamental requirement in the development of the model.

The connectivity between the elements is defined as "logical dependence" (Checkland & Scholes, 1990). This supports the view that the SSM conceptual models are intended to be models of concepts, rather than models of physical objects or events, because logical relations cannot exist between physical objects or events. Logical dependence is illustrated in the hypothetical conceptual model shown in figure 2. It shows that r is dependent on u and v.

There is a problem here because the elements of the SSM models are commands and generally accepted logics only operate on truth bearers (for a discussion of what is and is not a logic see Haack, 1978, and Grayling, 1990). Statements, or more strictly propositions, can be true or false and are, therefore, truth bearers. Commands can be neither true nor false and have no place in generally accepted logics. A logic of commands, an imperative logic, has been discussed by some authors (Haack, 1978) but Probert (1991, 1993) finds that an imperative logic is not enough to fulfil the role required of it in an SSM model.

This problem can be easily overcome by replacing the imperative phrases in the models with declarative phrases. Instead of putting "wash rice" we could put "the activity *wash rice* has occurred" and now the truth of this proposition could be said to be dependent on the truth of the proposition "the activity *obtain rice* has occurred". Or, more concisely, we could say "rice is washed" instead of "wash rice" and "rice is obtained" instead of "obtain rice". Figure 3 shows how the commands of figure 2 can be replaced by propositions.

2.3.2 The problem of insufficiency

Accounts of the logic of causation are in terms of necessary conditions, sufficient conditions and necessary and sufficient conditions (Copi, 1968; Mackie, 1980; Papineau, 1978; Taylor, 1963). Logical dependency, which is the only relation used in SSM models, is parallel to a necessary condition. If the truth of the statement "rice has been washed" is logically dependent on the truth of the statement "water has been obtained", then obtaining water will be a necessary condition of washing rice. However, the relation of logical dependency does not amount to sufficiency; obtaining water is not sufficient for washing rice.

In figures 2 and 3 if we say r is logically dependent on u and v we are saying the same thing as saying u and v are necessary for r, but this does not mean that u and v are sufficient for r. The logical way of expressing this is to say that r implies u and v. In the symbolism of the propositional calculus:

$r \rightarrow (u \,\&\, v)$

Here the truth of *r* allows us to infer the truth of *u* and *v*. However, the truth of *u* and *v* does not allow us to infer anything about *r*. In causal terms the fact that *r* happens means that *u* and *v* must have happened but the fact that *u* and *v* happen does not mean that *r* will happen. If we think of the arrows as representing implication, as they do in symbolic logic, then the arrows point the wrong way in SSM models. The upshot of this, in simple English, is that the fact that rice and water are obtained does not mean that the rice gets washed.

This entails that a physical system that is based on a model that contains only necessary conditions can never be guaranteed to work. It may work because the necessary conditions may in fact also be sufficient but it is also possible that they might not be.

This deficiency can easily be remedied by adding another condition that, in conjunction with the existing conditions, forms a set which is sufficient. The way this can be done is shown in figure 4. Here the set comprising *w* and *u* and *v* is sufficient for *r*. As each of these conditions (*w*, *u* and *v*) is also necessary for *r*, the set is a necessary and sufficient condition (N&S condition) of *r*.

In figure 4 the new elements corresponding to agents are introduced in order to create sets of sufficient conditions. The new element *w An agent is employed who will wash rice if rice and water are obtained* when combined with *u Rice is obtained* and with *v Water is obtained* is sufficient for *r Rice is washed.* That is, if we employ an agent who will wash rice if rice and water are obtained *and* we obtain rice and water *then* rice will be washed.

Agents could correspond to people, machines or, in the case of an information system, a computer program. Traditional SSM models are models of human activity systems and it is reasonable to think that implicitly the presence of a human agent has been assumed. While sets of sufficient conditions are required for all full explanations of causal sequences, agents are not. Figure 22 give a full causal account of measles without needing to introduce the notion of an agent or an agency.

It is not immediately obvious when agents need to be included in a full account of a causal sequence. Some man made systems such as cooking rice, making chair legs (figure 7) and painting fences (figure 10) seem to require them. Others such as the hospital system (figure 21) and the public relation project (figures 28 - 33) do not seem to need agents. It might be simply that if these models were expanded to sufficient depth the need for agents would become apparent.

Another factor is that in the cooking rice, making chair legs and painting fences examples the movement is towards a monitor and control system that would bring about a certain state of affairs. In the hospital, measles and public relations system the concern is more to develop a system that will correspond to an existing state of affairs; if detailed monitor and control systems were added to these then agents might be required.

However, there might also be a philosophical problem here. The notion of an agent or agency as it is used here has parallels to the ancient notion of causal power or causal efficacy. By this account state of affairs A could only cause state of affairs B by virtue of A's power to do so. Causal power was an ontological notion and today it tends to be regarded as an otiose metaphysical extra. However, Taylor (1963) recently resurrected this notion. A full discussion of Taylor's paper would be an unnecessary digression, but it does not appear to be obvious when the notion of agency needs to come in to an account of causation.

2.3.3 Showing different possibilities

The introduction of N&S conditions solves the problem of insufficiency but deficiencies in the model remain. Necessary conditions and N&S conditions show only one way in which the objective can be attained. For example, if we want to obtain equipment this could be accomplished by buying it or making it or borrowing it.

Wilson recognizes this and tries to accommodate it by making distinctions between *what* is to be done and *how* it is to be done. The *how* can be expanded in two ways: by showing greater detail (in effect, more necessary conditions) or by showing different possibilities.

Mingers (1990) has shown that the what/how distinction is not, in itself, capable of making the distinctions that are required. One problem is that the expansion of the model by taking it to a higher resolution level, which is similar to the way data flow diagrams are decomposed, will fail to make clear whether the model is being expanded in order to show greater detail or whether it is being expanded to show one of a number of possibilities.

Mingers suggest a qualified what/how distinction as the basis for the development of the two hierarchies, one to show greater detail and one to show different possibilities. There is, however, another way of introducing different possibilities into the models; this involves the introduction of a third logical connection.

2.3.4 Introducing SUN conditions

So far we have two types of logical connection:

A -> B. Which means A implies B. This corresponds to B being a necessary condition of A.

A <-> B. Which means A implies B and B implies A. This corresponds to B being an N&S condition of A, and, as a logical consequence, A being an N&S condition of B.

To this group we can add:

B -> A. Which means that B implies A, or that A is *implied by* B. This corresponds to B being a sufficient but unnecessary condition (SUN condition) of A. SUN conditions are indicated by broken lines in figure 5. (A notational device for SUN conditions is useful but not essential see section 11.3.1.2)

With solid lines to indicate necessary conditions and broken lines to indicate sufficient conditions we could indicate A being an N&S condition of B by two lines one, solid and one broken, going from A to B. However, if A is an N&S condition of B, then B will be an N&S condition of A; therefore, four lines would be required. To avoid the diagrams becoming to messy a double headed arrow will be introduced to stand for N&S conditions at a later stage, see figure 16.

It is self evident that if it is true that polished rice is obtained then it will be true that rice is obtained. We can, therefore, say that obtaining polished rice is a sufficient condition of obtaining rice, and that "polished rice is obtained" implies "rice is obtained". While the truth of "polished rice is obtained" is a sufficient condition of the truth of "rice is obtained" it is not a necessary condition because rice can be obtained without obtaining polished rice, in the case in point rice can be obtained by obtaining unpolished rice.

The SUN conditions for any event, or for the truth of any proposition, form a set. The occurrence of the event or the truth of the proposition does not entail that any individual member of the set obtains or is true; however, it does require that at least one member of the set obtains or is true. This means that if we know that u is true then one of c, d, e and f must be true, if this is not the case then the set of SUN conditions for u (the set comprising c, d, e and f) will not be exhaustive. If the model is not exhaustive then it cannot be universal and cannot account for every case. The way to make sure that a model is exhaustive is to make sure that each set of SUN conditions cover all possibilities.

In figure 5, *c* and *d* cover all the possibilities for *b*. That is, if polished rice is obtained then the rice that is obtained must be domestic or imported. There is no other possibility, therefore, *c* and *d* form an exhaustive set of SUN conditions for *b*.

The break down of *q*, in figure 5, has deliberately been left so that it is not exhaustive. It is reasonable to think that if rice can be cooked with borrowed equipment it can be cooked with stolen equipment. Therefore, "equipment is stolen" is a SUN condition of *q*. The easiest way to correct this is to include "equipment is stolen" as an additional element. However, in many cases the stake-holders would not want to consider the possibility of stealing being part of their system. Fortunately other solutions are possible. We can omit stealing from the model but still make it exhaustive by altering *p* from "rice is cooked" to "rice is cooked by legal means". In this way the models begins to become linguistically as well as logically dynamic.

A third possibility is to take it that "legal means" is part of the Universe-of Discourse for the system. That is, we can take it that the model is not intended to cover all possibilities but only legal possibilities. This limitation could be recorded by amending the root definition to include legality. This going back to modify the root definition following an inadequacy in the model would be undertaken as part of the iterative process. The interrogation of the stake-holders' concepts is a large part of what the model building is about.

With the inclusion of necessary, N&S conditions and SUN conditions conceptual models are capable of representing any conceivable cause and effect sequence. These types of model will have far greater scope than models which are based directly on past experience.

2.4 Applications of the models

2.4.1 Cause & effect diagrams

2.4.1.1 Ishikawa's diagrams

Cause and effect diagrams are closely identified with the work of Ishikawa (1986). His book on quality control devotes considerable space to the subject. Ishikawa's account of causation is inadequate in two ways. Firstly, he does not distinguish between necessary conditions and sufficient conditions. Secondly, he does not take logical possibilities into account.

For Ishikawa "a cause" is broken down into other causes and these in turn can be further broken down into yet other causes. At any given point, therefore, it is difficult to understand what Ishikawa means when he uses the word "cause". He could be meaning a necessary condition, a sufficient condition, or a necessary and sufficient condition.

Figure 6 is taken from one of Ishikawa's cause and effect diagrams. The effect, delicious rice, is represented at the end of the main arrow. Leading into this are four arrows labelled "Pretreatment (washing)", "Raw Materials (rice)", "Equipment (cooker)", and "Second treatment (steaming)". It appears that these are meant to represent necessary conditions but it is not clear if they are meant to represent a set that is sufficient.

At the lowest level the diagram seems to list SUN conditions. The the upper right part of the diagram would seem to be saying that obtaining rice from Thailand or obtaining rice from China are SUN conditions of obtaining rice from foreign countries. However, these SUN condition could not be considered an exhaustive set unless one thought it would be impossible to make delicious rice from, say, American rice or Indian rice.

Figure 6 is one of the most comprehensive of the Ishikawa diagrams. In practice, cause and effect analysis in the Ishikawa tradition sometimes gives little more than an ordered sequence of events that have been involved in a production or distribution system (for example, see Jones & Clark, 1990). Ishikawa's research method is confined to the study of the past performance of a system. Like all such work its scope is very limited. It tells us very little about what *could* happen nor, in a rapidly changing environment, is it likely to tell us what *will* happen.

The greatest advantage of logico-linguistic models over the Ishikawa type is the fact that they can cover all logical possibilities. This brings us back to the point that a conceptual model need never have a smaller scope than an empirically based model. This is because anything that is known empirically is conceivable and can, therefore, be included in a conceptual model. By contrast some things that are conceivable can never be known empirically.

2.4.1.2 Fault trees

Ishikawa's method developed in the context of quality control and many of his diagrams are directed at finding the causes of faults. A causal account of a fault merely reverses the logic of a desired state of affairs. If we want to achieve X, and Y is a necessary condition of X, then *not* Y will be sufficient for *not* X; to put it another way, Y will be sufficient for a fault. By the same reasoning, if Z is a sufficient condition of X, then not Z will be a necessary condition of a fault. The use of Ishikawa's models in this context does not, therefore, avoid the difficulties raised above.

Fault tree analysis is a rigorous method of fault detection used in engineering. It employs flow diagrams containing input events, AND gates and OR gates. The occurrence or non-occurrence of an input event provides the equivalent of logical negation. Given this, a fault tree is capable of representing the full range of causal conditions. For example $p \rightarrow q$ could be expressed as *(p AND q) OR (NOT p AND q) OR (NOT p AND NOT q)*. However, fault tree analysis *presupposes* a comprehensive system description (Barlow & Lambert, 1975) whereas conceptual modeling and cause and effect diagrams are meant to *provide* a systems description.

2.4.2 A logical account of efficiency

2.4.2.1 The problem of efficiency in SSM

Checkland & Scholes (1990) indicate that most systems should be accompanied by three measures of performance: efficacy (E1), efficiency (E2) and effectiveness (E3).

The criterion for efficacy will tell us whether the desired effect has occurred or not. In the case of figure 5 this will amount to whether *p* is true or not. If *p* is false we know that *t* or *r* or *q* or *s* must be false, and if *r* is false we know that *w* or *u* or *v* must be false. From this an algorithm can be formulated that will find the faults in a system and take remedial action. E1, therefore, has a useful role.

Effectiveness (E3) is the measure of whether the system meets a longer term aim. In the case of our example this might be to enjoy a good meal. The criterion for E1 would be *is rice cooked?* If the criterion for E1 is met it remains an open question whether the criterion for E3 is met. The fact that rice has been cooked does not entail that we enjoy a good meal. Better systems might be to fry potatoes, go to a restaurant or to hire a caterer. E3, therefore, also has a useful role.

Problems arise when we come to consider efficiency. Checkland & Scholes define efficiency as "amount of output divided by amount of resources used". There is a difficulty here because SSM models consist entirely of necessary conditions. If a system is to work, no necessary condition can be left out. This means that any system that consists entirely of necessary conditions can operate in only one way. Which leaves the question: what is the criterion for efficiency meant to measure?

The introduction of SUN conditions into the models can provide the role for the criterion of efficiency and, thereby, solve this problem. We can say that the system is efficient if the only SUN conditions that are true are those that meet the predetermined criterion that we have selected as E2. The criterion for efficiency can select the optimal

SUN condition, or set of SUN conditions, needed to achieve E1. As a consequence it will minimize unnecessary conditions and thereby eliminate redundancy.

It is worth pointing out that there is nothing in the logic of SSM that requires that the criterion for efficiency be quantitative. In the *cook rice* example the criterion could be palatability. We can take the account of efficiency in terms of SUN conditions to be a logical concept of efficiency. As such it can be contrasted with the mathematical concept.

2.4.2.2 A logical concept of efficiency

Figure 7 gives a model of a system to make chair legs. The input for the system is square lengths of wood and the output is round lengths with holes provided for cross piece joints.

The model serves to illustrate how time can be introduced into the models as well as showing how causes of efficiency can be identified. If the final event, *p*, takes place at *T*, then *q*, *s*, *r* and *w* must take place at *T* minus 1. If *q*, *s*, *r* and *w* take place at *T* minus 1, then *u*, *b*, *a*, *c* and *v* must take place at *T* minus 2.

There are two ways in which this system can operate. One is by drilling the holes in the square lengths and then making the lengths round on the lathe; this way invokes the *w* SUN condition. The other is to make the lengths round and then drill the holes; this way invokes the *s* SUN condition.

Given a criterion for efficiency as the number of lengths produced per day, it is quite likely that one method will conform to the criterion better than the other. It might be that difficulties in positioning a round piece of wood prior to drilling make the *s* route less productive. Alternatively a hole in the length might interfere with the smooth operation of the lathe, making the *w* route less productive.

To determine which of the possibilities is, in fact, the most efficient, would require experiment or monitoring the system in real world application. However, the important thing here is that this question of efficiency was recognized without comparison with other systems as would be required for a mathematical account of efficiency. The other important thing is that the parameters of efficiency here have been recognized without acquaintance with any real world chair leg making system. This suggests applications in a green field situation.

2.4.2.3 Mathematical ideas of efficiency

The mathematical idea of efficiency takes a system to consist of inputs, a black box and outputs. A system *A* will be taken to be more efficient than system *B* if the ratio of outputs to inputs is higher in *A* than in *B*. Data Envelopment Analysis is more sophisticated but the black box remains and, for the purposes of this discussion, it can be treated as the same as the simple input/output account.

While the mathematical concept will help to identify efficiency it does not identify the cause of efficiency. Take two systems, *A* and *B*, with comparable inputs and outputs but in which *A* is determined to be more efficient. There are two possibilities as to the cause: it could be external to the systems, or it could be internal to the systems.

If the cause is a factor that is external to the system then it would seem that the cause is really an input, but perhaps one that has been overlooked. Let us suppose that *A* and *B* are farms in which the inputs are seed, fertilizer, manpower and equipment, and the output is grain. Let us suppose *A* does better then *B* because *A* is situated in a place where the weather is better than it is at the location of *B*. We do not want to say that the weather is internal to the systems as far as efficiency is concerned. This is because our concept of efficiency, unlike the concept of productivity, requires that we can make changes that can improve it. So, as we cannot change the weather we add it to the list of inputs. If the weather was the only cause of the low productivity of *B*, then *B* should now have the same efficiency rating as *A*.

Given this we must conclude that any true cause of efficiency or inefficiency is internal to the system. But if the cause of efficiency is internal an analysis of inputs and outputs cannot locate it. To identify the cause of efficiency or inefficiency of two systems would require a comparison of their internal configuration.
Logico-linguistic conceptual models are one of the ways in which a system's internal configuration can be described.

2.4.3 Automated monitor & control

2.4.3.1 Sets of sufficient conditions

SSM conceptual models developed into propositional logic expressing causal sequences could be used for the design of automated monitor and control systems. The first stage is to express exhaustive sets of SUN conditions, that is SUN conditions that cover all possibilities. The example that will be used is taken from *Soft Systems Methodology in Action* (Checkland & Scholes, 1990) and is reproduced as figure 8. The model is derived from the following Root Definition: "A householder-owned and manned system to paint a fence, by conventional hand painting, in keeping with the overall decoration scheme of the property, in order to enhance the visual appearance of the property".

As in all traditional SSM models the arrows in the model are meant to represent necessary conditions, which means that *5 Paint the fence* is contingent upon *4 Obtain materials*, but they do not represent sufficient conditions. The fact that the activity *Obtain materials* is performed does not imply that the activity *Paint the fence* is performed. If we think about applying this model to a real world situation there is no guarantee that it will work. It is quite possible (and with a tedious task like painting a fence, even likely) that the materials are obtained but the householder never gets round to doing the painting.

Figure 9 shows a way in which sufficient conditions for painting a fence can be presented. This is by adding three new elements which stand for the notion of human agents which are implicit in the fence painting model. The elements standing for the agent designate a required role. This is specified in greater detail than in figures 5 or 7. The element *5a A agent is employed who will make 5 true if 4 is true* is detailed in term of two necessary conditions (*5a* and *5c*) that are jointly sufficient. The conjunction of *5a* and *5b*, therefore, form a set that is necessary and sufficient. One of these, 5c, is that the agent is competent, this is defined in terms of the system. That is, the fence will be painted if the agent is competent, if the fence is not painted then the agent cannot be competent. This enables control action to be specified: if the materials have been obtained but the fence does not get painted then the agent cannot be competent and the agent should be changed. However, the system might be badly configured and be such that no agent could possibly accomplish the task. In this case 5b, *5 is possible if 4 is true*, will be false and the system should be changed. The precise configuration of the monitor and control action will depend on the particular circumstances. The number of possible agents will depend on the environment. Quite often it will be enormous in which case some cut off point will need to be determined. For example there may be thousands of painters available for employment, in this case we could build in the rule that if the agent has been changed, say five times, and the fence has still not been painted then the system should be changed.

The logic of the model can now be explained. With the central part of the fence-painting model converted into propositional form it can be expressed in the propositional calculus:

$$(5 \rightarrow 4) \;\&\; ((4 \rightarrow (2 \;\&\; 3)) \;\&\; (2 \rightarrow 1)$$

In this formula the original numbers from the Checkland & Scholes model are retained and stand for propositions. This makes it easier to see how the formula is derived but breaks with the convention in logic that propositions are denoted by lower case letters. The use of numbers in the formulas has also caused difficulties because some people have been reading mathematical or temporal significance into them. In subsequent examples only letters will be used.

Translating the implementation system into propositional form and adding the sufficient conditions gives figure 10. Part of this model can be expressed in the following formula from which some interesting deductions can be made. These could be used for system diagnostics:

$(5 \leftrightarrow (4 \,\&\, 5a)) \,\&\, (5a \leftrightarrow (5b \,\&\, 5c))$

Therefore: $-5 \rightarrow (-4 \vee -5a)$

and: $-5a \rightarrow (-5b \vee -5c)$

This means that if 5 is false (if the fence is *not* painted) then 4 must be false or 5a must be false (the materials are *not* obtained or an agent who will paint the fence if the materials are obtained is *not* employed). Also that if 5a is false (an agent who will paint the fence if the materials are obtained is *not* employed) then 5b must be false or 5c must be false (that it is *not* possible for an agent to paint the fence or that the agent is *not* competent).

This makes it easy to track a fault in the system. If the measure of performance for efficacy is not met (if the fence is not painted), it must be because materials have not been obtained or the agent has not performed the task. If the agent has not performed the task this must be because the agent is incompetent or because the system is unworkable in its present form.

The same process occurs one stage back. From figure 10 we know:

(4 <-> (2 & 3 & 4a)) & (4a <-> (4b & 4c))

Therefore: -4 -> (-2 v -3 v -4a)

and: -4a -> (-4b v -4c)

Again we have the same form of inference. If materials have not been obtained, it must be because the colour to paint the fence has not been decided or because the scope of the fence painting task has not been described or because the agent has not performed the task.

In any model elements standing for agents can be attached to any element for which a necessary condition has been specified (any element which has an arrow going into it). These new elements are shown by the broken lines in figure 5 (the fact that the lines are broken has no logical significance). Those elements which do not have arrows leading into them (those for which necessary conditions are not given) do not have elements standing for agents attached to them; this is because the agents will appear when the model is taken to a higher resolution level.

2.4.3.2 Structured English for control

From figure 10 it can be seen that there are three types of way in which the system can go wrong, these are:

Type One: Problems of supply from outside the system. In this case it would be 1 or 2 not being true.

Type Two: Problems with the competence of agents. In this case it would be 5c, 4c or 3c not being true.

Type Three: Problems because it is impossible for the agents to do their job, in this case it would be 5b, 4b or 3b not being true.

Given this it is easy to specify the control action. If the problem is of Type One, then the supply sub-system should be changed. If the problem is of Type Two, then the agent should be changed. If the problem is of Type Three, then the operating system should be changed, with Type Three problems the model is unworkable. Type Three problems would require a new conceptual model and a return to CATWOE and the root definition.

The following gives a simple algorithm, in structured English, for control action based on the monitoring of efficacy.

```
(1) IF 5
      THEN stop
      ELSE IF not 4
             THEN goto (2)
             ELSE IF not 5c
                  THEN change agent
                  ELSE change system

(2) IF not 3
      THEN goto (3)
      ELSE IF not 2
             THEN change sub-system
             ELSE IF not 4c
                    THEN change agent
                    ELSE change system

(3) IF not 1
      THEN change sub-system
      ELSE IF not 3c
             THEN change agent
             ELSE change system
```

An obvious objection to what has been said above is that obtaining materials and employing an agent might mean that fence painting has begun but it does not mean that the fence is painted. This problem can be overcome if we build in temporal references.

Say it takes six hours to paint the fence. We can amend element 5 to "the fence is painted at *t*", element 4 can be amended to "materials are obtained at *t - 6 hrs*", and element 5a can be amended to "an agent is employed who will make it true that the fence is painted at *t* if materials have been obtained at *t - 6 hrs*".

2.4.3.3 Organizational mapping

The method of developing a conceptual model given above represents a considerable refinement of the organizational mapping techniques developed by Wilson (1984, pp. 149-256). If elements 1, 3c, 2, 4c and 5c can be mapped onto a real world situation we can be sure that the fence will be painted or that the system is unworkable.

As Wilson's models are based on necessary, but not sufficient, conditions there is no similar guarantee. We could find that when one of these models maps onto the real world the system works or we could find that it does not.

The method also allows for greater detail in mapping. For example, the revised models could be mapped onto individual workers in a production line. With Wilson's models this cannot be done, they are confined to mapping boundaries and areas of responsibility; in effect they are concerned only with monitor and control functions such as those shown in figure 8.

The third advantage is that Wilson's method requires that the entire conceptual model (expanded to three or four higher resolution levels and sometimes comprising hundreds of activities) is mapped onto what is known about the entire organization (derived from organization charts plus field research), whereas the revised models could be mapped piece by piece.

To explain this, let us suppose that in our fence painting example the problem situation is that the fence is not getting painted. The first task is to produce the conceptual model (figure 10). The second task is to see if elements 4 and 5a can be mapped onto the real world situation. If element 4 cannot be mapped, i.e. if the materials are not obtained, then we try to map elements 3, 2 and 4a. If the words *change agent selection system* is substituted for *change agent* and the words *change Conceptual Model* substituted for *change system* then organizational mapping can follow the general lines of the algorithm given above.

2.5 Limitations of propositional logic

The logical analysis of SSM models at a propositional level can have uses in understanding and stimulating the SSM debate. Merali (1993) has described how, firstly, logically enhanced models can be used to encourage stake-holders to rethink the problem situation and secondly, how the consensus that is reached at the end of the debate can be expressed in a logical model.

Merali (1992) has also shown how logico-linguistic models at the level of propositional logic can be converted into analytic, that is conceptual, data flow diagrams. This opens the way for SSM modelling methods to be used in a solution of the structured methods type. However, while this might have some useful applications, the limited power of a structured methods solution and the limited power of the propositional calculus imposed severe limitations on the range of applications.

An obvious limitation is the absence of data analysis in a structured methods solution. It seems that propositional logic is inadequate to support data analysis and that the more powerful predicate logic is required. Logico-linguistic models based on propositional logic tend to be awkward to construct. The propositions the models need to refer to each other; for example, 5b in figure 9 needs to refer to 5 and to 4. This is inelegant and cumbersome and could easily lead to problems of circular reference in large models. This problem disappears when predicate logic is introduced.

The greatest shortcoming of both propositional logic and structured methods is that they cannot distinguish between the notional status of SSM models and status of real world facts. As was stressed in section 1.1.5 it is the notional status of the SSM models that makes then startlingly different and, therefore, interesting. The next chapter will seek to account for this special status in terms of Wittgenstein's language game theory. Chapter 4 will develop the logical tools needed to express it.

3 A WITTGENSTEINIAN PERSPECTIVE ON SSM

3.1 The context

3.1.1 Meaning and information system design

The aim of this chapter is to show how SSM conceptual models, which are notional, can play a role in the design of information systems intended to support real world activities. Attempts to explain this, especially Checkland's, have been in the area of the theory of knowledge (epistemology). They have tried to show how a notional model, systems idea or "holon" plays a role in the acquisition of knowledge. This chapter takes a new initiative and addresses the problem from the perspective of the theory of meaning. It is contended that meaning is the main problem here and that once this is sorted out many of the epistemological problems will be easily resolved.

A consideration of meaning was prompted by the logical problems discussed at the end of the last chapter. The foundation of any applied logic is inexorably tied up with meaning. The literature of logic and, to a certain extent, artificial intelligence abound with elaborate formalisms which have only a tenuous relation to practical problems. An account of meaning will provide the grounding for logic and information system design. It will be argued that the SSM conceptual model building process is a type of Wittgensteinian language game in which a new language is created to describe the problem situation. Given this the model building will express meaning that can be used as the basis for the logic needed for an information system design.

Connecting Wittgenstein with information system design is not new but other writers have tended to emphasis Wittgenstein's work on the contextual aspect of language (see Hanseth, 1991). Wittgenstein's emphasis on the study of language as it is *used* largely gave rise to speech act theory. Speech act theory is now employed in modelling for IS design in the SAMPO approach (Auramaki et al, 1988, forthcomming). Here language game theory is employed to help the analyst understand the meaning of terms used in the context of on-going activities. In this chapter a different role will be considered, that of a language game as preliminary to new activities in an existing situation or new activities in a new situation.

In the Investigations Wittgenstein draws a connection between language and games especially chess. The main point is that both are constituted by their rules. But the rules of language like the rules of a game can be invented.

> To invent a language could mean an instrument for a particular purpose on the basis of the laws of nature (or consistently with them); but it also has another sense, analogous to that in which we speak of the invention of a game. (Philosophical Investigations, 492)

Wittgenstein did not see any fundamental difference between the rules of a so-called natural language, such as English, and the rules of a so-called artificial language, such as predicate logic. For Wittgenstein the rules that govern language were not just *approximate* to logic but were of the same order.

> ... For it will then also become clear what can lead us (and did lead me) to think that if anyone utters a sentence and *means* or *understands* it he is operating a calculus according to definite rules. (Original emphasis, Philosophical Investigations, 81)

However, although the rules of a language can be invented, the terms of a language cannot refer to a purely subjective state of mind. It was a fundamental principle of Wittgenstein's later philosophy that the creation of meaning was a public event. This is the aspect of his theory that is relevant to SSM. The iterative debate in which the conceptual models are built is a public event that creates meaning.

Contrary to the later Wittgenstein, subjectivist theories of meaning assert that the determination of meaning is essentially private. This needs to be qualified as the word "subjective" in information systems literature is apt to do more to confuse than to clarify. We find that "subjective" is sometimes use to stand for "private" and sometimes used to stand for "individual". That meaning is subjective in the sense of "individual" is hardly in dispute. It is obvious that different individuals mean different things by the same utterance and that to say that a term is meaningful is to imply the existence of at least some individual people.

The contention that meaning is subjective in the sense of "private" is called "the ideational theory" by Grayling. He describes it as:

> The theory...that language is an instrument for reporting thought, and thought consists of successions of ideas in consciousness. Ideas are private; only I have access to my own thoughts. Therefore to communicate our ideas to each other we need a system of intersubjectively available sounds and marks, so connected to ideas that the proper use of them by one person will arouse the appropriate ideas in the other person's mind. Accordingly what a word means is the idea with which it is regularly connected. (Grayling, 1990)

Ideational theories reached their most rigorous exposition in the theories of Logical Atomism and Logical Positivism which were propounded in the 1920s and 1930s. Wittgenstein's early work (Tractatus Logico-Philosophicus, 1922) was very close to these movements. However, in the 1940s Wittgenstein changed his position and attacked the ideational theory. The main instrument of this attack was the private language argument. In Anglo-American analytical philosophy this argument is generally regarded as having shown that ideational theories are false. However, they now seem to be appearing in writing relating to information system design.

3.1.2 Outline of the argument

Wittgenstein's argument against the ideational theory is an old one but quite complex. For those readers who are not familiar with the private language argument a summary set in its historical context is given in Section 3.2.

Section 3.3 expresses a concern that ideational theories are beginning to appear in work on the connection between meaning and IS design. The work of Stamper is briefly discussed because despite a good deal of similarity between Stamper and Wittgenstein they differ in important particulars. The contrast between the ideational theory and Wittgenstein's is described. A consideration of Wittgenstein's theory of truth shows that even formal systems can be regarded as language games. The section seeks to establish that it is the language game theory of meaning, and not the ideational theory, that is correct.

An argument for a Wittgensteinian account of SSM model building is given in Section 3.4. It is argued that one of the products of the SSM iterative debate is an agreement about how the problem situation should be described.

Understood as a language game the debate is not merely a mechanism for the analyst to learn what the clients think but creates an agreed framework of stipulative definitions. The section seeks to establish that Checkland's descriptions of the logical status of SSM models and the actual process of building the models are both compatible with the language game theory but not with the ideational theory.

Section 3.5 considers how the work in the previous sections can be developed toward a system for information system design. The shortcomings of the propositional calculus are described. The need to establish a system that will deal with the problem of mapping conceptual models on to the real world is pointed out.

3.2 The history of private languages

3.2.1 Logical Atomism

To find an irrefutable foundation for human understanding has been the ambition of countless philosophers throughout history. Bertrand Russell was no exception. In the early part of this century he began developing a set of ideas that became known as "Logical Atomism" (Russell 1918, 1924).

Russell began with the standard empiricist idea that all knowledge comes through the senses. His next move was to say that the only thing we can be certain of is sense experience. Knowledge is, he argued, built up entirely out of atomic units of sense experience. These units he called "sense data". The standard Platonic analysis of knowledge which is that it is *true, justified belief*. Russell was saying that only sense data justify belief and thereby turn it into knowledge. This was nothing particularly new, the sense datum theory can be traced back at least as far as John Stuart Mill.

Russell's next move was rather unusual. He went on to say that not only was knowledge built up out of sense data but that meaning was also built up out of sense data. This was quite profound because if sense data are required for meaning they are also required for the formulation of belief. For Russell it was not possible to even believe anything that was not based on sense data. This, therefore, was a unified theory of knowledge and meaning.

In the 1905 paper "On Denoting" Russell developed his theory of descriptions. This claimed that most names are in fact disguised descriptions. Names apparently refer to individuals but "On Denoting" hoped to show that names can be unpacked into logically equivalent descriptions which have sense but no individual reference. Russell (1918) went on to say that the only "logically proper names" have individual reference and these only refer to sense data.

The elegance of this theory was very appealing. Wittgenstein's *Tractatus* (1922) with its "picture theory of meaning" is firmly in this tradition. Rudolf Carnap and the Vienna Circle were working on similar ideas which were popularized in Britain by A. J. Ayer (1936) in his best selling book *Language, Truth and Logic*. More than any other text this book represents the views that people began to call "Logical Positivism".

3.2.2 Empiricism and phenomenalism

Taken independently of the theory of meaning, the sense datum theory is just one species of the philosophical position known as "phenomenalism". This comes from John Stuart Mill who held that objects were just "permanent possibilities of sensation" (Ayer, 1969, p. 224-5). The empiricist says that all knowledge *comes from what we have experience of*. The phenomenalist takes this further and says all knowledge *is made up of experiences*, for the phenomenalist what these experiences are *experiences of* is something we cannot know.

The position known as "phenomenology" ends up being similar to phenomenalism in its account of the external world. However, phenomenologists, such as Edmund Husserl, get there by a different route. This is the route of rationalism which claims that knowledge is *a priori*, a result of thought rather than a result of experience.

The trouble with phenomenalists and phenomenologists it that they board up the window to the outside world leaving the subject completely alone.

3.2.3 Language games

By the 1940s Wittgenstein had changed his mind completely about the nature of language. In the *Philosophical Investigations*, which was not published until 1953, he produced an argument that was fatal to Logical Atomism, Logical Positivism and many of the ideas in his own *Tractatus*. This became known as "the private language argument". The private language argument shows that it is not possible for a language to refer to objects that only one person can, as a matter of logic, know about. Sense data are logically private because only one person can know his own sense data.

The private language argument is a complex one and the exposition here will be limited to an outline. Kenny (1973) considers that the crux of the argument is that the terms of a private language could not be defined. He identifies three prongs to the attack. First, it contends that a private object, a sense datum such as a pain, cannot be ostensively defined. That is, a person cannot merely fix his attention on a sensation and name it "so and so".

> ...what does it mean to say that he has 'named his pain'? - How has he done this naming of pain?! And whatever he did, what was its purpose? - When one says "He gave a name to his sensation" one forgets that a great deal of stage-setting in the language is presupposed if the mere act of naming is to make sense. (Investigations, 257)

Secondly, a private sensation cannot be defined in terms of a previous sensation.

> We are supposing that I wish to justify my calling a private sensation 'S' by appealing to a mental table in which memory-samples of private objects of various kinds are listed in correlation with symbols... To make use of such a table one must call up the right memory-sample: e.g. I must make sure to call up the memory-sample that belongs alongside 'S' and not the one that belongs alongside 'T'. But as this table exists only in the imagination, there can be no real looking up to see which sample goes with 'S', i.e. remembering what 'S' means. But this is precisely what the table was meant to confirm. In other words the memory of the meaning of 'S' is being used to confirm itself. (Kenny, 1973, pp. 192-3)

Thirdly, a private sensation cannot be defined in terms of public events.

> Let us now imagine a use for the entry of the sign "S" in my diary. I discover that whenever I have a particular sensation a monometer shews that my blood-pressure rises. So I shall be able to say that my blood-pressure is rising without using any apparatus. This is a useful result. And now it seems quite indifferent whether I have recognized the sensation *right* or not. Let us suppose that I regularly identify it wrong, it does not matter in the least. And that alone shews that the hypothesis that I make a mistake is a mere show. (We as it were turned a knob which looked as if it could be used to turn on some part of the machine; but it was a mere ornament, not connected with the mechanism at all.). (Investigations, 270)

To replace the idea of language as something based on reference to logically private objects and events, Wittgenstein developed the idea of language as consisting essentially of rules. In the *Investigations* the notion of a language game is developed. A language is like a game. You cannot play the game if you don't obey the rules but the rules are no more that an agreement among the putative players about how to play the game. There are many games that you can play and new ones are being made up all the time. For the later Wittgenstein language is public, and the references in any language are learned from publicly observable objects and events.

3.3 Theories of meaning

3.3.1 Connotation and denotation

The word "meaning" has a wide range of usage in the English language and only part of that range will be relevant to the arguments in this chapter. We will be concerned with meaning only in so far as it is about the understanding of the elements in a language. These are for example, words, terms, sentences and statements. We shall not be concerned with the following types of usage: *meaningful* relations between lovers, a *meaningful* action, what a red sky at night *means*.

Meaning can be broken down into sense and reference. Frege made this distinction with the terms "sinn" and "bedeutung" (1892), John Stuart Mill with the terms "connotation" and "denotation" (1843) and the terms "intension" and "extension" are also used to make the same distinction. This distinction is now included in standard logic textbooks. Proper names have denotation but no connotation, but general terms have both. "A general or class term *denotes* the objects to which it may correctly be applied...The properties possessed by all of the objects in a term's extension are called the *intension* or *connotation* of that term" (Copi, 1968). A simpler way of putting this is that sense and connotation are concerned with the relation of terms to other terms, while reference and denotation is concerned with the relation between terms and objects or events.

3.3.2 The ideational theory revived

The ideational theory has an intuitive plausibility for modern Western people, but as Grayling points out "The word "idea" entered into ordinary English usage only a few centuries ago, until which time it had been strictly a philosophical term of art." It is not until the theory is worked through that its initial plausibility is seen to be chimerical. One would expect the ideational theory to appeal to a computer scientist who is just beginning to come to grips with the wider aspects of information system design. When the word "subjective" appears in the context of meaning in the information systems literature the author is often taking a tacit ideational stance.

Nevertheless, a tendency towards the ideational position is not confined to writers unfamiliar with the theory of meaning. For example, a fairly recent paper on semiotics by Ronald Stamper (1987) seems, at first glance, very close to Wittgenstein. Stamper emphasizes agreement and context as essential to the understanding of meaning. He also explicitly disavows the ideational theory (the position he calls "psychologism"). However, a close attention to the paper indicates a tacit ideationalism. This is evident when we consider what he has to say about Tarski's theory of truth. This theory can be expressed very crudely as the theory that a statement is true if and only if it refers to an actual state of affairs. Thus:

"Snow is white" is true if and only if snow is white.

Stamper attacks this theory because he thinks it makes two big assumptions:

> There is a definite, independently existing world (for without it we cannot resolve semantic questions this way). There clearly is, or equally clearly is not, a correspondence between a sentence and any world of which it might be stated. The first is again the naive belief in one objective reality..." (Stamper, 1987)

The first point to notice about this is that Stamper is overstating the theory, when he says that the correspondence must be clear, if he means that all claims to a correspondence are incorrigible. The theory makes no such assumptions. For example if all the swans I have encountered are white I will be justified in believing that "swans are white" and if I do this I will hold that "swans are white" is true. Later if I encounter a black swan I will realize that my earlier belief, although justified at that time, was in fact false. The main object of the correspondence theory is to show that justified belief and truth are not always the same thing.

Be this as it may, the main point is that it is difficult to see how Stamper can avoid an ideational stance if he rejects the notion of an independently existing world. This is because we are forced to ask what "swans are white" refers to if there is no independently existing world. If there is no "independent" world then, presumably, there must be some sort of dependent world, and what sort of world is this? A passage at the beginning of the paper might help to clarify Stamper's position:

> Meanings express personal views of reality. When there is a firmly established consensus, and only then, we can pretend that meanings are independent of people." (Stamper, 1987)

What Stamper seems to be implying here is that the subject alone determines meaning - that public languages are a sort of Esperanto built up out of private languages. From this it follows that essentially "swans are white" refers to something subjective - one of numerous personal realities. This is certainly ideational. If we agree with Wittgenstein then it is clear that Stamper has the boot on the wrong foot. Language is public and refers primarily to public events. References to subjective sensations are derived from a public language. I don't make up my own word for my pains and then translate it into English. I learn the use of the English word "pain" by observing public events and then apply the word to my own pains. This does not prevent people from giving different meanings to the same utterance or symbol, this is because the same utterance or symbol is used in different language games. There can be symbols that mean a certain thing in computer jargon but mean something completely different in prison slang. This is because computer scientists and prisoners play different language games. Which language game is being played at a particular time will depend on the history of the players and the context in which it is being played.

Stamper appears to assume that meaning must be entirely objective or entirely subjective. He reasons that meaning cannot be entirely objective because words do not have meanings in themselves, and, therefore, meaning must be entirely subjective. But his assumption is wrong. Meaning is public and as such it is dependent upon the existence of at least two knowing individuals *and* dependent upon the existence of independent and observable objects and events.

3.3.3 Truth and rules

In the proceeding section a discussion of the correspondence theory of truth was used to draw out Stamper's tacit ideational position. A complication must now be added because Wittgenstein did not accept correspondence theories of truth either. However, his reasons were quite different from Stamper's. The later Wittgenstein subscribed to the redundancy theory of truth. This is the theory that saying that "All swans are white is true" or saying that "It is a fact that all swans are white" is the same as saying "All swans are white".

An objection to the redundancy theory of truth is that without a correspondence to facts sentences cannot form truth-functional compounds. Kripke gives Wittgenstein's answer as:

> We *call* something a proposition, and hence true or false, when in our language we apply the calculus of truth functions to it. That is, it is just a primitive part of our language game..." (Kripke, 1982).

This reveals an important part of Wittgenstein's thought. For Wittgenstein formal systems such as mathematics and logic are language games. For Wittgenstein reference was not possible outside a rule based language game. A new game can be devised and played by a group of people agreeing to a set of rules. In the same way a language game will produce rules and these rules can be formalized. In the following sections it is argued that the SSM conceptual model building process is, in part, a language game. As such it offers a viable alternative to attempts to design information systems on the basis of an ideational theory of meaning.

Before proceeding I shall summarize the difference between Stamper and Wittgenstein. Stamper objects to the correspondence theory because he thinks there is no independently existing world. For Wittgenstein there must be an independently existing world because language cannot refer to a logically private world. Wittgenstein's objection to the correspondence theory is that it is part of a language game not something that stands over and above all language games. However, whether Wittgenstein's objection fires or not depends on how we take the correspondence theory. If we do not take it as a supreme principle, but take it as something to distinguish between justified belief and knowledge within a certain system of definitions, then I do not think Wittgenstein would object to it.

3.4 SSM and Wittgenstein

3.4.1 SSM and subjectivity

It seems that in information systems circles the term "interpretivist" is now being used to denote methods that imply that a social situation is open to more than one interpretation, while the term "positivist" is used to denote methods that imply that there is only one valid account of a social situation. In this sense SSM is interpretivist, as opposed to positivist, in its account of social events. This might lead one to believe that it must be ideational in its account of meaning, but this does not follow. An ideational account of meaning entails an interpretivist account of social events and this is why they tend to be found together. However, a rejection of the ideational account of meaning does not entail that there is only one valid account of a social event. The language game theory of meaning is compatible with the idea that there are a number of equally valid ways of describing a social event.

3.4.2 SSM as a language game

Checkland has always been adamant that SSM conceptual models are neither true nor false, nor are they correct or incorrect. They are not intended to be a representation of a real world state of affairs. However, if they are neither true nor false they cannot be representations of anything. They cannot, for example, be a representation of the stake-holders' ideas. Nor can they be a representation of what the stake-holders mean by something.

It is this fact that suggests that conceptual model building can be explained, at least partially, as a type of Wittgensteinian language game. If we take it that the stake-holders and the facilitator are playing a game in which a new language is created to describe the problem situation, then the validity of the model does not require that it has a truth value. Taken as a language game the building of a conceptual model is a public event in which the stake-holders come to an agreement about the terms that can be used to describe the problem. The model, therefore, has the logical status of an agreement and agreements are neither true nor false. A more formal explanation is that the finalized conceptual model (the one that marks the end point of the iterative debate) is a definition of a desirable state of affairs. Here the definition is a stipulative definition, and once again by the standard account (see Robinson, 1954), stipulative definitions are neither true nor false. (This is an acceptable account if we limit ourselves to non-modal logics. Later, modal logics, which distinguish two types of truth, will be introduced. When this is done we will be able to say that stipulative definitions are "logically true" but factually neither true nor false. This is still compatible with Checkland as he can be understood as saying that conceptual models are not factually true or false.)

This mode of explanation is not open to ideationalists because for them meaning ultimately resides with ideas in the subject's consciousness. From this it would follow that a conceptual model would only be meaningful in so far as it represented ideas in a subjective consciousness. If this were the case it would be true to say that the model was meaningful if it did in fact represent the subject's ideas and it would be false to say it was meaningful if it did not. Thus for ideationalists a putatively meaningful conceptual model would be representational and, therefore, true or false.

Although they cannot be true or false putative conceptual models can be valid or invalid. Validity is concerned with consistency within a set of rules rather than with representation. We can talk of a valid agreement and by this we mean that the agreement conforms to certain rules and regulations for making agreements. A valid agreement in British law is one that conforms to British law on agreements. With conceptual models validity will consist of conformity to the rules of conceptual model building. For example, a rule for SSM conceptual model building is that every bubble must have an arrow going into it or an arrow coming out of it.

The rules of SSM model building are a higher order language game. Before the stake-holders can start building a SSM model relevant to the problem situation they must accept the rules of SSM model building. If they do not want to accept these rules, that's fine - they can do something else; but they cannot build an SSM model *and* refuse to accept the rules.

SSM rules are part of another language game and subject to a yet higher set of rules. These rules include the basic laws of logic such as the law of non-self-contradiction. Once again it is up to the stake-holders whether they want to play this game. If they do not, and they are happy with self-contradiction, that's fine. But in this case they are hardly likely to want a computerized information system.

This account is compatible with much of what is written about the models but also with the practice of SSM. One of the thrusts behind the development of SSM was the concern to develop a way of dealing with messy problem situations, and its ability to do so remains its principal attraction. One of the reasons that messy problems arise in organizations is because the stake-holders cannot, literally, understand each other. The stake-holders often have different functions, come from different backgrounds and have been educated in different disciplines. They rarely share the same vocabulary. The conceptual model building process enables a common vocabulary to be built up.

3.4.3 Modal logic and truth in SSM

To regard conceptual models as being valid but neither true nor false is one solution. However, a more elegant and potentially useful solution to the problem is offered by modal logic. Modal logics distinguish between necessarily true statements and contingently true statements. Necessarily true statements or formulas are prefixed by the "*L*" while contingently true statements are prefixed by the "*M*" modal operator. Another way of making this distinction, which avoids confusion with the notion of causal necessity, is to say that the "*L*" operator stands for what is logically true and the "*M*" stands for what is factually true.

Logically true statements are true either by virtue of the sense of the words used in them "all bachelors are unmarried" or because they follow logically from other necessarily true statements. Although necessary statements cannot be false they can be meaningless because they can be constructed in such a way that they refer to nothing. "All unicorns are unicorns" is necessarily true but meaningless because unicorns do not exist.

Factual statements cannot be necessarily true because nothing in the real world is true because of the way words are defined. Therefore, it is always logically possible for any factual statements to be false, and, therefore, factual statements are always contingent.

Modal logics are used to deal with counterfactual conditions. Such statements as "if Jones had listened to his accountant he would be rich today" are considered to be true in circumstances where Jones did not, in fact, listen to his accountant and where Jones is not, in fact, rich today. Modal logicians account for this by saying that in one possible world Jones did listen to his accountant and in that world he is rich. This analysis is known as "possible world semantics". The modal operator "*L*" is used to denote what has to be true in all possible worlds, while the operator "*M*" is used to denote what is false in at least one possible world.

If we accept that conceptual models are stipulative definitions then they cannot be factually true. If they are to be truth functional and play a part in a logical calculus they must, therefore, be necessarily true. We can now return to Checkland & Scholes remark, quoted in Chapter 1, about validity:

> Since the model does not purport to be a description *of* part of the real world, merely a holon relevant to debating perceptions of the real world, adequacy or validity cannot be checked against the real world. Such models are not, in fact, "valid" or "invalid", only technically defensible or indefensible. (Checkland & Scholes, p. 41, original italics).

What is being stated here is that the model is not factually true or false. In modal logic we can accept this but still be able to say that the model is logically true. Modal logic, therefore, captures Checkland's sentiment very well.

3.5 Conclusions about meaning and information

3.5.1 Meaningful foundations

Logical Atomism and Logical Positivism are the most forceful attempts at a ideational account of meaning. The leaders of these movements all abandoned this position in their later writing. The ideational account of meaning takes us back to the beginnings of Logical Atomism. There is, therefore, a danger that the information systems literature will start to rerun a futile debate that began in the 20s and finished in the 60s. This will not be necessary if due attention is given to the legacy of Wittgenstein.

Understood as a language game the SSM iterative debate provides both a firm theoretical foundation and a powerful practical tool for the development of information systems. Computer systems are rule bound and formal. It has been argued above that there are rules implicit in SSM conceptual models, these rules can be developed and formalized in logico-linguistic models. This opens the way for the rigorous development of computerized information systems that will be meaningful not only to professional analysts and designers but also to the people who use and interact with the system.

3.5.2 Improved logic

This section has shown how conceptual models, understood as language games, could be formally expressed in modal logic. This overcomes one of the difficulties that were found in the use of propositional logic in Chapter 2. A second shortcoming is that propositional logic does not distinguish between universal statements and particular statements. This needs to be done, tacitly or otherwise, in every information system. For this the predicate calculus is required.

4 UNIVERSALS & PARTICULARS

4.1 The need for predicate logic

Chapters 2 and 3 indicated a general need for the use of predicate and modal logics in the analysis of SSM models. While Chapter 3 was concerned with the overall status of the models the present chapter continues the work of chapter 2 by examining the technical details of the models. Attention to the technical details give a second line of argument for the use of predicate and modal logics.

The Chapter begins with a discussion of Probert's (1991, 1993) challenge to the fundamental basis of SSM Modelling. Wilson and Checkland & Scholes (1990) claim that their conceptual models show logical dependencies. Probert argues that their models cannot be logical in any sense of the word. This argument can be answered by the introduction of predicate logic with its distinction between universal and particular statements. However, making this distinction reveals a number of important facts about conceptual models. The first fact is that the conceptual models produced by Wilson and Checkland in their Soft Systems Methodology (SSM) are "logical" in a quite different way from the way in which the Multiview models are "logical". These two senses of "logical" can be distinguish by using modal logic.

The universal/particular distinction has implications for information system design. The rules for Knowledge Based Systems can be derived directly from universals. The argument, therefore, turns full circle. Objections to the logic of conceptual models prompts a deeper analysis, the deeper analysis shows how the models can reveal a logical structure suitable for information system design.

In spite of this, there are ambiguities about the status of SSM models. This is brought out in section 4.3 which considers the origins of universals. Three types of universal are identified: inductive hypotheses, value statements and definitions. In section 4.4 an argument is made, complementary to the language game theory, that the SSM models are definitional. This creates problems in determining that a given model has any relation to the real world because to establish this would seem to require inductive hypotheses. Section 4.4 indicates that Multiview models, by contrast, are collections of inductive hypotheses. The section argues that Multiview is faced with a problem that is the opposite of SSM: a hypothesis can only be formulated in a language, a language requires definitions and the Multiview model building does not generate definitions.

The chapter concludes by suggesting that a new type of model could be constructed. The new models would include definitions and inductive hypotheses thereby combining the advantages of the SSM method with those of Multiview. A further feature could be the inclusion of value statements. This would make the models logically comprehensive as well as adding a new dimension to the model building process.

4.2 The "universal" solution

4.2.1 Probert's Problem

The word "logic" has come to mean a variety of things. The Collins English Dictionary gives seven definitions, only two of these will be relevant here: 1. the branch of philosophy concerned with analyzing the patterns of reasoning by which a conclusion is drawn from a set of premises, without reference to meaning or context. ... 6. the relationship and interdependency of a series of events, facts etc.

The conceptual models in SSM are represented as words contained in bubbles which are joined by arrows. The arrows are intended to show a relationships of "logical contingency". Thus in figure 11 a bubble containing the words "Discharge patient" is connected by an arrow to another bubble containing the words "Apply treatment". A vital point here is that this could function on one of, at least, two levels. The arrows could be intended to show logical contingency between the expressions in the bubbles (first level) or logical contingency between real world events that correspond to these expressions (second level).

If the logical contingency is intended to be at the second level then this will be consistent with the definition of logic in sense 6. This is exactly what "logic" means in the Multiview conceptual models: "use arrows to join the activities that are logically connected to each other by information, energy, material or other dependency..." (Avison & Wood-Harper, 1990, p. 60).

This is quite different from the account of conceptual models given by Wilson and Checkland. The language game explanation of SSM conceptual models not only fits well with mainstream writing but also allows us to describe the models as being "logical" in sense 1. This in turn allows the models to be expressed as being logically true in modal logic. Therefore, provisionally at least, we can take it that the SSM models are not logical in sense 6. and that the Multiview conceptual models are a different type of thing.

Probert's argues that the arrows in the SSM models cannot be logical in sense 1. because the contents of the bubbles are imperatives, and logical relationships in sense 1. can only hold between declaratives. "Discharge patient" is a command and standard logics only operate on statements or propositions. This point was anticipated in chapter 2 by the suggestion that the commands could be converted into parallel statements. Thus, "Discharge patient" could be converted into "The command *Discharge the patient* has been obeyed" or simply "The patient has been discharged". Similarly we can substitute "Treatment has been applied" for

"Apply treatment", and "Treatment has been prescribed" for "Prescribe treatment". This gives Figure 12 which can be expressed in the propositional calculus as:

$(t \rightarrow s) \ \& \ (s \rightarrow r)$.

But Probert still does not find this satisfactory. His argument (1993, p. 240) can be paraphrased as follows "*the patient has been discharged* does not logically entail *treatment has been applied.*" His point depends on what we take $t \rightarrow s$ to mean. There is a certain ambiguity here that can only be resolved by using predicate logic.

4.2.2 The hidden premise

If we add a universal statement to the two statements given above, the problem is resolved:

Major Premise: All cases where a patient has been discharged are cases where treatment has been applied.

Minor Premise: A patient has been discharged

Conclusion: Treatment has been applied (by *Modus Ponens*)

Here we have the classic syllogism in which the major premise is a universal, the minor premise is a particular, and the particular conclusion follows by *Modus Ponens*.

Probert's Problem is that the conclusion here cannot logically follow from the minor premise without the major premise. This means that the model shown in figure 12 is a logically contingent argument. It is not, as it stands, a logically valid argument. The universal given above is a hidden premise in the argument that the figure represents.

It can be noted that this will usually be the case with figures such as figure 12. The statements in these types of figure are always particulars. It it is unusual that a particular conclusion can be drawn from particular premises. Normally a particular will be deduced from a particular premise and a universal premise. An exception is simple conjunction, we can deduce "Socrates is a tall man" from the particular premises "Socrates is a man" and "Socrates is tall".

Although figure 12 is not logically valid it is not false, and it can be made into a logically valid argument by adding universals. This can be done using the predicate calculus. This involves the use of quantifiers. "A" will be used as the universal quantifier and "E" as the existential quantifier. Normally these are printed upside down but typographical constrains prevent this here. We can express figure 12 in predicate logic as follows:

Domain: people who go to hospital

Tx: x is a patient who is discharged

Sx: x is a patient who is treated

Rx: x is a patient for whom treatment is prescribed

1 Prem (Ax) Tx -> Sx

2 Prem (Ex) Tx

3 (Ex) Sx From 1 and 2 by *Modus Ponens*

4 Prem (Ax) Sx -> Rx

5 (Ex) Sx From 3

6 (Ex) Rx From 4 and 5 by *Modus Ponens*

This can be rendered in English as follows:

1 For all patients, if a patient has been discharged then that patient has had treatment.

2 At least one patient has been discharged.

3 At least one patient has had treatment.

4 For all patients, if a patient has had treatment then that patient has had a treatment prescribed.

5 At least one patient has had treatment.

6 At least one patient has had treatment prescribed.

The first three lines here repeat the syllogism given above. To this is added, at 4, another universal premise. Given this we can deduce 6. This is illustrated in figure 13 where the elements from figure 12 have been expressed in the predicate calculus and the two universals have been added; the arrows have been included only to show the general flow of the argument.

The example given here is a simple one because the domain contains only one type of object and the predicates are all one placed predicates. However, this does not effect the argument as the numerous distinct objects and *n*-placed predicates can be dealt with in essentially the same way.

It is significant that with the universals included in this way, the arrows from figure 12 are no longer necessary. The universals replace the arrows. It is no longer necessary to state (*p* -> *s*) & (*s* -> *r*) because this is contained in the two universals. The whole of figure 11, and any other conceptual model, can be expressed entirely in terms of universals.

4.2.3 SSM models as universals

An interesting fact about universals is that they do not, in themselves, commit us to the existence of anything. (Ax) Tx -> Sx does not imply that anything exists. It could just as easily represent "All unicorns eat ambrosia", which does not imply the existence of unicorns or ambrosia. It is not until we add a particular statement that there is any commitment to existence. That is why (Ex) is called the existential quantifier. In the argument above, existential commitment begins with (Ex) Tx, the fact that there is at least one patient who is discharged. Once existence has been introduced existential consequences follow, such as (Ex) Sx, the fact that at least one patient has had treatment.

Expressing SSM conceptual models in universals ties in well with the idea discussed above: that the models are elaborate definitions not models of what exists, necessarily, in the real world. The predicate calculus highlights this distinction and shows that the model will only map on to the real world if a set of particular statements are true. This prompts epistemological questions that will be fully addressed in a later section.

4.2.4 Knowledge Based Systems

A direct information systems application is now apparent. The type of formula given above has an immediate counterpart in Prolog programming. The universals correspond directly to Prolog "rules". Lines 1 and 4, above, would be:

```
has_treatment (X) :- is_discharged (X).

has_treatment_prescribed (X) :- has_treatment (X).
```

the particulars would correspond to Prolog "facts". However, there could not be a direct counterpart to the existential quantifier. A Prolog program requires a value for the x in

(Ax); to put it more precisely, there must be an *instantiation* instead of the object variable in a Prolog program.

In the example, this would be satisfied by naming a person, say Socrates, who is discharged:

is_discharged (socrates).

Given this Prolog "fact", the program will return the answer "socrates" when asked who has had treatment prescribed. This, therefore, is a rudimentary Knowledge Based System. It can also be noted that there are parallels between universals and field structure, and between particulars and records, in traditional data base design. There are, therefore, good indications that a relational data base design can be derived from these predicate calculus formulas. How to do this will be shown in section 7.5.

4.3 Three types of universal

4.3.1 Inductive hypotheses

Expressing SSM conceptual models in terms of universals escapes logical objections and thereby solves the immediate problem. Nevertheless, difficulties remain because the universals are, as they stand, contingent. All we have done is swap a contingent model in the propositional calculus for a collection of contingent universals. The status of the model will, therefore, depend upon where these universals come from. The most natural answer would be that they are "factual statements", that is, inductive hypotheses about the real world and based on real world experience. But in this case the models would not be notional. However, this conclusion can be avoided because the universals in the model need not be inductive hypotheses. We can distinguish two other types of universal, these are value statements and definitions.

4.3.2 Value statements

Value statements include statements about personal tastes, such as "all of Shakespeare's plays are rubbish", and moral statements, such as "everyone ought to give to charity". Value statements can be distinguished from factual statements by a number of logical and epistemological properties. Evidence can be used to support or falsify factual statements but not value judgements. "All swans are white" is given supportive evidence by the observation of more and more white swans, and it falsified by the observation of one black swan. There is no evidence for "everyone ought to give to charity" nor can it be falsified empirically.

Value statements connote a certain form of behavior. If Icabod believes that "everyone ought to give to charity" then Icabod will approve of charitable acts. Although they cannot be falsified empirically, two value statements can be shown to be incompatible with each other when they connote contradictory behavior. Factual statements do not connote any form of behavior.

From the logical point of view it is plausible to construct conceptual models entirely out of value statements. We can imagine what this would look like. With the universals "All people who drop litter are bad" and "All bad people should be punished", and an instantiation, "Icabod drops litter", we can draw the conclusion: "Icabod should be punished".

Value statements alone, will not, of course, account for the models that are, in fact, produced in SSM. There is no way that "All cases where a patient has been discharged are cases where treatment has been applied" could be construed as a value statement.

SSM prides itself on being able to deal with the human aspect of a problem situation. It is, therefore, surprising to find that value statements have a very small role in the building of conceptual models. Value statements are usually implicit in the criterion for effectiveness and they can sometimes, but not always, be found in the Weltanschauung part of CATWOE. More recently (Checkland & Scholes, 1990) *Ethicality* (is T a moral thing to do) and *Elegance* (is T aesthetically pleasing) have appeared as additional measures of performance making a total of five Es. However, people have complex systems of values and few people ever share exactly the same set of values. SSM has made no attempt to model values and show how a number of different sets can be accommodated into a consensual view of what is ethical or elegant.

Apart from measures of performance values rarely appear. Despite the fact that the models are constructed in the language of imperatives these almost always turn out to be practical rather that value ridden imperatives. They are of the form "You should turn left if you want to get to the station" rather than "You should give to charity if you want to be good".

Another crucial difference between value and factual statements is that value statements are not reducible to factual statements and factual statements are not reducible to value statements. What this means is that we cannot derive factual statements from value statements. If we are to draw a factual conclusion we must have at least one factual premise, factual conclusions cannot be drawn from value statements alone. The same is true of value statements, a conclusion that expresses a value cannot be derived from purely factual premises. This point is summed up in the dictum "you cannot derive *ought* from *is*". Given this and SSM's anti-reductionist stance, it is even more surprising that value statement have such a small role to play in the models.

4.3.3 Definitions

If we accept that SSM conceptual models are not intended, in any straight forward way, to be models of things that are in the real world, then we are forced to the conclusion that the universals must be definitions.

A distinction is made between definitions intended to establish an existing meaning, *descriptive* definitions, and definitions giving a proposed meaning for the future, *stipulative* or *prescriptive* definitions.

Taking the SSM universals to be descriptive definitions has an initial plausibility. In this case the conceptual models would not be models of the real world but models of a language used to describe the real world. This could account for a lot of what happens in practice. Organizations tend to develop their own languages. The process of SSM conceptual model building could be taken to be a process whereby the stake-holders describe how this language is used. But if this were all that was going on the process would be quite simple and there would be no need for a lengthy iterative debate about the model.

Taking the SSM universals to be stipulative definitions is much more plausible and can be grounded in Wittgenstein's language game theory. Given this Figure 11 can be interpreted as a set of rules for the use of a language within a particular organization. The universals given in section 4.2.3 could be expressed as rules:

Rule 1: Nothing is to be described as "a discharged patient" unless it is preceded by something that can be described as "an application of treatment".

Rule 2: Nothing is to be described as "an application of treatment" unless it is preceded by something that can be described as "a prescription of treatment".

This account of conceptual models in terms of stipulative definitions fits well with the main thrust of SSM which is to address unstructured problems. Unstructured problems come about not because of a lack of structure *in the real world* but because of a lack of structure *in descriptions of the real world*. The creation of a cohesive set of definitions can provide the structure.

For example, if we want to find out if all Christians know the Bible, then the main methodological problem is going to be deciding what Christians are (People who say they are? People who goes to church regularly?) and what is meant by "knows the Bible" (All of it? Most of it? Some of it?). Once these things have been decided, collecting the real world data will be, methodologically, fairly simple.

4.4 Universals and the status of SSM models

4.4.1 Implication and entailment

Modal logic distinguishes two types of implication. These are material implication and strict implication or entailment (some logicians might hold that "strict implication" and "entailment" are not exactly the same thing, but for the purposes of this thesis the distinction is irrelevant). The difference between the two types of implication can be made using the two modal operators "*L*" and "*M*". Suppose that "all cats are animals" is true as a matter of definition. Then it is not possible to find a cat that is not an animal. In modal logic we would say that x being a cat strictly entails x being an animal. In symbols:

Cx: x is a cat

Ax: x is an animal

$L\ (Ax)\ Cx \rightarrow Ax$

Suppose that "All cats like milk" is true as a matter of fact. Then in at it is at least possible, but it may never happen, that a cat can be found that does not like milk. In modal logic we would say that x being a cat materially implies x liking milk. In symbols:

Kx: x likes milk

M (Ax) Cx -> Kx

4.4.2 Process definitions

This brings us back to Probert. If the conceptual models are essentially causal in nature then material implication is the strongest relation we can use to express them. If conceptual models are essentially definitional, as is contended in this chapter, then the stronger relation of entailment can be used.

The model represented in figure 11 looks like a cause and effect diagram and there is nothing in its logical expression that indicates that it is not. However, the manner of its construction shows that it is not a representation of a process of cause and effect but that it is a process definition. Most things are defined by their qualities. A chair can be defined as an object with a seat, a back support and more than two legs. Other things are defined by the process of their production. Whiskey is a spirit distilled from fermented malted grain. This means that if you take some grain, malt it, ferment it and then distill it you end up with whiskey - no matter what it tastes like. By the same token something that has the same taste, alcohol content and colour as whiskey is not a whiskey unless it is produced by the defining process. For example, in Thailand there is a popular liquor called "Mehkong" which is made as a substitute for Scotch. Originally Mehkong was made from rice, which is a grain, and Mehkong was, correctly, called a whiskey. These days it is made from molasses, which is not a grain, and so Mehkong is now described as a "liqueur" despite the fact that it has the same alcoholic content and looks and tastes the same as it always did.

Given that figure 12 is a process definition we can express the universals in figure 13 as relations of entailment:

Prem 1. L (Ax) Tx -> Sx

Prem 2. L (Ax) Sx -> Rx

The way is now clear for the construction of modal models. In Chapter 6 an enhanced logico-linguistic models, comprising entailment and excluding material implication, will be constructed. This is represented in figure 19.

4.4.3 Holons and modal models

Checkland & Scholes (1990) use the term "holon" to denote a system of thought. As such a holon can be distinguished from a system in the real world. A similar distinction can be drawn with regard to logico-linguistic models, such as figure 19, which comprise only logically necessary relations. Such a logico-linguistic Model is an extended definition and as such need not have any correspondence with the real world; the model could just as easily be that of the family tree of a Greek God as anything in the real world. A second important point about the notion of a "holon" is that there is no single holon that is correct in regard to a given situation. There can be a number of equally valid holons relating to the same situation. The same is true of logico-linguistic models; the same situation could be described using a different set of definitions.

There is also a similarity with axiomatic systems here. The mere fact that an axiomatic system has been formulated is no guarantee that it has any correspondence with the real world. We also find that the same system, such as the propositional calculus, can be formulated using different sets of axioms.

Expanded into logico-linguistic models the SSM conceptual modeling method can be a way of producing axiomatic systems. This stands to be very useful because one of the problems with axiomatic systems is that there is no logical reason for anyone to accept them. If there are reasons for accepting a statement then that statement must be some form of inference not an axiom. Generally it is said that axioms are self evident, but this is just another way of saying that they are accepted without reason. Admittedly some axiomatic systems, such as arithmetic, seem to be very useful whereas others do not. Nevertheless, before an axiomatic system can be shown to be useful it must be accepted, if only tentatively, and there is no reason to do this. SSM solves this problem pragmatically; as the stake-holders make up their own axioms the question of their acceptability does not arise. However, the question of their usefulness does arise, this will be addressed in Chapters 6 and 8.

4.4.4 The problem of reference

This chapter has been concerned with meaning *qua* sense (connotation, intension) with the meaning of terms in the context of other terms. It has not dealt with the question of how we can establish that a term refers to an existing state of affairs. This is a fundamental issue that is formulated in different ways. In the language of SSM the question is: how can we tell when a conceptual model maps on to the real world? In the theory of truth it is: how do we establish that a statement corresponds to a fact? In Wittgenstein's philosophy it would be: how do we know when a language game is useful? For Hofstadter (1980) it would be: how can we establish that an axiomatic system is isomorphic?

Essentially all these questions are asking: how can something that has been made up help us to understand and describe something that has not been made up.

4.4.5 SSM models and the real world

If the universals that constitute the SSM models are definitions, then it follows that the models will be analytic rather than synthetic. That is, if they are true they are true by the meaning of the words alone. If this is the case, there is nothing in the conceptual model building process that guarantees that the models can refer anything that exists or could exist. There is nothing that prevents them from including references to unicorns, Greek gods and flying pigs. Obviously, models that contain these types of reference cannot be used as a basis for information system design or for organizational restructuring.

Similar problems will arise for those people who consider that the *only* point in building a conceptual model is to change peoples' thinking. This is because a change in thinking can only be useful if it contains a reference, directly or indirectly, to an actual or potential real world state of affairs. Pragmatic and verificationist theories of meaning hold that without such a reference a change in thinking is not just useless but is, literally, a meaningless notion (see, for example, Ayer, 1946). The same can be said about values; if a change in values does not indicate a change in behavior in response to some actual or potential event in the real world, then it is pointless to say that there has been any change in values.

Establishing how an analytic system, such as arithmetic, maps on to the real world is the subject of complex and contentious theory. In the case of a conceptual model, such as that represented in figure 13, we would need to establish an instantiation for the object variable (Ex) Tx, i.e. that Socrates, or some other person, is discharged. Having established that Socrates is discharged, we can establish deductively that Socrates has had a treatment prescribed; this is true by definition. But if it is true by definition it cannot be true that Socrates is discharged and false that Socrates has had treatment prescribed. Therefore, in order to be sure that Socrates is, in fact, discharged we must be sure that he has had a treatment prescribed. But if we must be sure that Socrates has had a treatment prescribed before we can be sure that Socrates is discharged, then the deduction that Socrates has had a treatment prescribed tells us nothing new.

Problems of this order are the basis of the claim by some empiricists that tautologies tell us nothing about the real world. However, this would appear to be false because arithmetic is an analytic system, true by definition and a tautology, but arithmetic appears to tell us a lot about the real world.

This vicious circle can be avoided if there is an independent criterion for a patient being discharged, that is, a criterion that is not a definition. Suppose the completion of Form PQ7 is such a criterion. The relation between Socrates is discharged and Form PQ7 has been completed for Socrates will be a contingent relation. Let us further suppose that there is a similar criterion for a patient having a treatment prescribed, say, the completion of Form RX5. Now, we find that Form PQ7 has been completed for Socrates from this we infer, contingently, that Socrates is discharged; from this we deduce that Socrates has had a treatment prescribed; and from this we infer, contingently, that Form RX5 has been completed for Socrates. More detailed examples will be given in later chapters.

From this we can see how an analytic system has proven useful. It has allowed us to infer one contingent event from another, events that might otherwise not have been connected. But a stronger case than this can be made. It can be argued that definitions are not only useful for contingent inferences, they are logically necessary (see section 4.5.1 below).

4.4.6 The essential problem for SSM

The essential problem for SSM is that there is no logical reason why the stake-holders should come up with conceptual models (a set of definitions) that map on to the real world. Connected to this is the fact that SSM has no way to determine whether or not these do or do not map on.

It could be argued that the real world contingency is introduced at a later stage. In Wilson's method the real world seems to begin to enter when information inputs and output between activities are identified. However, it is not altogether clear whether these are meant to be notional information input/outputs between notional activities, or real world information input/outputs between real world activities. In either case there is still a problem.

If the information input/outputs are notional we still have to establish that they can map on to the real world. If they are real world information input/outputs between real world activities, where did the real world activities come from? How was it established that the notional activities from the Conceptual Model map on to real world activities?

4.5 Universals and Multiview models

4.5.1 Definitions & inductive hypotheses

The importance of establishing definitions for universals will be readily apparent when it is realized that there is no intrinsic way of distinguishing between definitions and inductive hypotheses or a fool-proof intrinsic way of distinguishing between definitions and value statements. Given that a certain universal is not a definition we can tell whether it is a value statement or an inductive hypotheses by certain key words that indicate values rather than objective facts about the real world. These include "should", "ought", "good", "bad", "nice", "nasty", etc. There is no set of words that can identify a definition.

Today "all men are mortal" would be considered an inductive hypothesis by most people. Most people would be likely to say that men are mortal because it has been observed that every man has died before, say, his 200th birthday. But for the Greeks "all men are mortal" was part of the definition of a man. The Greeks thought that some men-like beings lived for ever, but these were not "men" they were Gods. For us, "immortal men" is meaningful, it stands for a class that happens to be empty; but for the Greeks "immortal men" was a contradiction in terms.

Value statements entail certain forms of behavior. From the use of the key value words in a given utterance a certain form of behavior will normally, but not always, follow. If Icabod says "all Christian are good people" then, if his utterance was sincere, we would expect Icabod to approve of Christians and act appropriately; if this is the case then Icabod's utterance was a value statement. However, Icabod might have made the utterance sincerely yet disapprove of Christians, we can imagine that Icabod prides himself on being a bad person; in this case the utterance was not a statement of Icabod's values but part of Icabod's definition of the words "Christian" and "good".

A distinction can be made between intensive and extensive definition. An intensive definition gives the sense (connotation) of the *definiendum*. An extensive definition gives the reference (denotation) of the *definiendum*. In terms of classes an intensive definition will provide a criterion of class inclusion whereas an extensive definition will list all the members of the class. Thus, an intensive definition of "a human limb" would be *any jointed appendage on the human body*, an extensive definition would be *an arm or a leg*.

An argument that definition is logically prior to inductive hypotheses can now be put forward. Empirical evidence of class inclusion require that the class is defined independently of that evidence. For example, if we say that all panthers are black then, if this is to be an empirical statement, there must be defining criteria for panthers that are independent of their colour. If being black is one of the defining criteria for panthers then "all panthers are black" must be analytic and cannot, therefore, be empirical.

As a matter of fact, being black *is* a defining criterion for panthers. "Panther" is just the word for a black leopard. So, to say that "panthers are black" is just to say that "black leopards are black" and this *cannot* be established empirically. As it is logically impossible to observe a black leopard that is not black, observation could never falsify the statement "black leopards are black"; as observation can never falsify the statement, observation cannot provide inductive evidence for the statement either.

If a term has been given an intensive definition we can establish the extension of the term empirically. Thus, if we intensively define "human limb" as any jointed appendage on the human body, then it can be established empirically that all human limbs are arms or legs. Likewise, if a term has been given an extensive definition we can establish the intention of the term empirically. If we extensively define "human limb" as an arm or a leg then it can be established empirically that all human limbs are jointed appendages on the human body.

This distinction between definitions inductive hypotheses and the fact that English grammar does not distinguish them is one that Wittgenstein made using the terms "criteria" and "symptoms".

> The fluctuation in grammar between criteria and symptoms makes it look as though there were nothing at all but symptoms. We say, for example: "Experience teaches that there is rain when the barometer falls, but it also teaches that there is rain when we have certain sensations of wet and cold, or such-and-such visual impressions." In defence of this one says that these sense-impressions can deceive us. But here one fails to reflect the fact that the false appearance is precisely one of rain is founded on a definition. (Philosophical Investigations, 354)

What he is saying here is that we can only call something a false sensation of rain if we have defining criteria of rain that are independent of sensation e.g. in terms of barometric pressure (it cannot be raining if the pressure is very high, therefore, if the pressure *is* high any sensation that is normally indicative of rain must be deceptive).

4.5.2 Constraints on the Multiview model

Having answered the immediate logical problems facing the SSM model by a somewhat tortuous route, it is appropriate to point out that the Multiview model is not as simple as it might seem. At first glance the Multiview model seems to be a generalized model based on observation and as such theoretically unproblematic. On closer examination the model involves considerable logico-linguistic difficulties.

Figure 14 is a Multiview conceptual model taken from a case study for a Distance Learning Unit. The large arrows represent flows of physical things, the small arrows represent information flows between the subsystems. If this was a model of an existing Distance Learning Unit it would not be problematic, nor would it be interesting. It would just be a generalized version of a materials flow diagram and a conventional data flow diagram. However, in this particular case the Distance Learning Unit did not yet exist. The Conceptual Model was, according to Avison & Wood-Harper, *derived* from a root definition. This root definition was:

> A system owned by the Manpower Services Commission and operated by the Paintmakers Association in collaboration with the Polytechnic of the South Bank's Distance Learning Unit, to provide courses to increase technical skills and knowledge for suitably qualified and interested parties, that will be of value to the

industry, whilst meeting the approval of the Business and Technical Education Council, and in a manner that is both efficient and financially viable.

(Avison & Wood-Harper, 1990)

We can express the double headed arrow between *Administration System* and *Course Exposition System* in figure 14 as "there must be a mutual flow of information between an Administration System and a Course Exposition System". How could this be *derived* from the root definition?

As with the SSM model there must be a hidden premise. This would be "Whenever there are courses to increase technical skills etc. there will be an Administration System and a Course Exposition System and a mutual flow of information between them". This is, of course, a universal. We can now ask: where does it come from? As Multiview statements are at the second level, referred to in section 4.2.1, it must be an inductive hypothesis based on the observation of other courses.

As we saw above, inductive hypotheses cannot be separated from definitions. The universal here would seem to specifying part of the extension of the term "courses to increase technical skills etc." this assumes that there is an intensive definition of the term. "Technical skill" needs to be defined outside of the full extension of the term "courses to provide technical skills etc.". "Technical skill" could not be defined in terms of passing the exam, for example; because, in this case, the only thing that "technical skill" would mean would be that the exam was passed. "Technical skill" needs to be defined in terms of some external factor such as the ability to paint a fence (presuming that painting a fence is not part of the extension of the word "course").

4.5.3 The limitations of Multiview

There are two possible ways in which Multiview can work in practice. One is where the stake-holders already have a well defined common language. The other is where the definitions that an information system requires develop informally during Multiview systems analysis.

The danger with Multiview is that in any given application the common language may fail to exist and may fail to develop. This danger is compounded by the fact that Multiview does not have the means to determine whether the common language is there or not. The danger can be avoided if definitions were included in the Multiview model building process.

4.6 A merger of models

SSM conceptual models pose a dilemma. An adequate account of their logic requires the use of universals. If these universals are definitions, as Checkland and Wilson seem to suggest, then we are faced with the problem of how they can map on to the real world. If the universals are inductive hypotheses, as Avison and Wood-Harper seem to suggest, then we can have problems if there is not an agreed language to bind the inductive hypotheses together.

A solution will be presented in the next chapter. This consists of building a logically enhanced model of the Checkland and Wilson type, a logico-linguistic model, and then adding elements from a Multiview type model to produce an empirical model. This method maintains the spirit and *modus operandi* of SSM while introducing logical rigor. The method enables modal logic to be introduced as a structured part of model building process.

In the remainder of the thesis universals that express value statements will not be considered again. In modal logic a value statement can probably be treated as logically true because they cannot be falsified by real world facts. The neglect of value statements is not a result of lack of interest but because the interplay of definitions and inductive hypotheses is so complex that a detailed account will take up the remaining space.

5 REAL WORLD MAPPING

5.1 The mapping problem

This chapter will address the real world mapping problem that was discussed in Chapter 1. It will use the modal logic that was introduced in Chapter 4 to show how SSM conceptual models can be mapped on to the real world. The narrative will be based around an extremely abstract example and this, hopefully, will illustrate the abstract nature of the problem. The method will be repeated in Chapter 6 where the example will be more concrete.

Section 4.5 showed that real world mapping was not an immediate problem with Multiview models. Nor is it an immediate problem for Checkland. With Checkland it is the building of the model as much as the model itself that solves the organizational problem. Checkland is concerned with changing the way the stakeholders think about their problem rather than with producing a detailed plan for a problem solution. In practice the problem sometimes simply disappears during the model building process. Other outcomes might be the identification of an organizational *raison d'etre*, or the definition of a new role for a department. In the context of information systems Checkland's models tend to be models of how to set up an information system or models of the organizational context of the information system, they are not models of the information system itself.

Wilson takes the models further and uses them as the basis for information system design. With Wilson the conceptual model is developed until it becomes the information system. While Checkland's models are fairly well accepted as being a useful front end to IS design, Wilson's work is more contentious. There are those that think it is not possible to produce an information system design from a conceptual model (see Mingers, 1992). It is not difficult to see why they are concerned. Wilson starts with a notional model and ends up with a system that handles information about real world entities such as stock in a warehouse (Wilson, 1984, pp. 195 - 208). It is common sense that our notions do not always correspond to reality and that what is desired cannot always be achieved.

This is not to say that any information system designed by Wilson, or designed using Wilson's method, will inevitably fall into error. In practice Wilson has designed a lot of perfectly good systems. The question is whether Wilson's method will always produce good systems in circumstances where good systems can be built. Wilson's successful results might just be a result of the fact that real world mapping has not been a particular problem with the systems he has built. These would be circumstances where the stakeholders conception of a desirable system happens to coincide with what can be achieved. This idea becomes credible when we consider how Wilson came to undertake his projects.

Wilson and Checkland worked together in the Department of Systems at Lancaster University for more than a decade. The main impetus to their research was the programme of action research. It is interesting to note that Checkland tended to take on the more unstructured, abstract and human centered research projects while Wilson took on the more structured, concrete and practical ones. The stake-holders in Wilson's projects were likely to be "down to earth" types whose concepts tend to coincide with the physically possible.

Another factor is that, compared to Checkland, Wilson's work has been more concerned with practice than with theory. It is quite possible that attention to the real world mapping problem has been made tacitly, perhaps even unconsciously, during Wilson's projects.

This and the following chapter will be concerned to make explicit what must be implicit in any method of information system design that is based on a conceptual model and is intended to operate with information about the real world.

5.2 The logico-linguistic model

When the idea of Logico-linguistic modelling was presented at United Kingdom System Society seminar on the subject of SSM and IS (Proceedings in *Systemist* Vol 14, No 3, Aug.

1992) Prof. Checkland suggested that one be constructed on the basis of the Gor Tonking model. This model is given along with the root definition and CATWOE in figure 15.

This is particularly appropriate to the real world mapping problem because:

> The model...cannot possibly include unjustified real-world knowledge since it is, deliberately, an RD [root definition] without meaning. It is included to show how a defensible logical structure for a model can be created from an RD, even though the RD does not refer to the everyday world." (Checkland 1989).

Another way of putting this is to say that the model has sense but no reference. If the language game explanation is accepted all conceptual models are like the Gor Tonking models in this respect; at the time of their completion (stage 4 in the learning cycle) they will have sense but reference will have to be established later (stage 5). The SSM learning cycle (Checkland, 1989) is shown in figure 18.

The real world mapping problem is concerned with whether the model can have reference. In the following the Gor Tonking model will be used to illustrate a procedure to determine whether a model can have reference. We will not assume that the model does not have reference but imagine that we are facilitators building a model for stake-holders who use a slang that we do not as yet understand. In the example we will try to map the Gor Tonking model on to a system to re-spray scratched cars. In principle the facilitator could come up with the mapping idea on his own initiative, in practice it would be better to have the stake-holders construct the entire model as they will know how their ideas are intended to map on to the real world.

To first step towards real world mapping is convert and expand the conceptual model into a Logico-linguistic model. As in previous chapters the commands are turned into statements. Thus the command "Tonk gors meeting the criterion" is replaced with "Gors meeting criteria gog are tonked". "Ascertain which gors meet criteria gog" is replaced with "Gors meeting criteria gog are selected".

Figure 16 is a partial Logico-linguistic development of just one arrow from the Checkland model - the arrow between element 5. and element 6. in figure 15. Two new statements "*r* Tonking materials are available" and "*s* A competent agent is employed to tonk gors" are introduced to make a set of sufficient conditions for "*p* Gors meeting the criterion are tonked". Two new logical devices are introduced. These are the biconditional and the AND containing box. The AND containing box stands for conjunction. In the figure it means "q and r and s". The double headed arrow stands for the biconditional or mutual implication. In the propositional calculus the figure will be expressed as follows:

$(q \& r \& s) <-> p$

It comprises two relations of implication:

$(q \& r \& s) -> p$

and

$p -> (q \& r \& s)$

This can be expressed in English as "Gors meeting criteria gog are tonked if and only if Gors meeting criteria gog are selected and Tonking materials are available and a competent agent is employed to tonk gors".

If we accept the language game interpretation the gor tonking model will be a set of stipulative definitions. Given this and the distinction between material implication and entailment made in the last chapter it is clear that both "$(q \& r \& s) -> p$" and "$p -> (q \& r \& s)$" are cases of entailment rather than material implication and can be expressed in modal logic as:

$L\ (q \& r \& s) -> p$

and

$L\ p -> (q \& r \& s)$

When two proposition entail each other we can say that each is strictly equivalent to the other. This can be compared to material equivalence where two propositions that materially imply each other. So "(q & r & s) <-> p" must be a case of strict equivalence, and can be expressed as:

L (q & r & s) <-> p

Modal operators can now be included in the diagrammatic technique. Figure 17 has an "L" symbol next to the double headed arrow to indicate that this is a case of strict equivalence rather than material equivalence. In layman's terms it means that the relation between "q & r & s" and "p" is one of definition rather than a relation that holds as a matter of fact.

With this the logico-linguistic conceptual model is complete for this limited example. It should, however, be noted that in chapter 6 two more logical connectives will be introduced into the logico-linguistic models in order to accommodate a more complex example.

5.3 Fixed and incorrigible rules

As was pointed out in section 4.3.3, logico-linguistic models can be understood as a set of linguistic rules. They have the status of being logically true and they are incorrigible by real world facts. We must now consider whether these rules can be changed at all.

Let us consider two people, Icabod and Isabel, are playing chess. They agree that the game has become boring and they think they would have more fun if the rules were changed. They decide to change the rules for the movement of the Queen. This is that the Queen should be able to move to any unoccupied square on the board in any single move. The two people then play a game with the new rule. In these circumstances does it make sense to say that they are playing chess? It would make more sense to say that they are playing a new game - we might call this "QChess". The interesting parallel with conceptual modelling here is that chess playing readers will easily be able to imagine what a game of QChess will be like, even though a game of QChess has never been played.

Now suppose that every chess player in the world decides to adopt the new rule after December. Will we be justified in saying, after December, that they play chess? We might be; but a better way to describe this situation would be to say "the games played before December and the games played after were both called "chess" but they were different games".

There is, therefore, a good case for saying that the rules of a language cannot be change without it becoming a different language game.

In itself it is not that important whether we say it is the same game with different rules or whether it is a new game with new rules. What is important is how the change of rules or new game comes about. The game does not change itself, changes come about because the players decide to play a different game. This is quite different from changes in a particular game of chess where changes happen because pieces are moved in accordance with the rules. A change in the rules of chess cannot be brought about by the rules of chess. A change in the rule of chess is of higher order than a game of chess.

A logico-linguistic model has the same status as the rules of chess. It lists the rules of a language game that the stake-holders have agreed to play. It is not intended to be a representation of the way the stake-holders speak.

It might be that the physical world is governed by immutable laws. But if it is these laws are quite different from the rules of chess. They are not made up by people. If they exist they are discovered by people. The rules of chess can tell us only about games being played by people who obey the rules of chess and even this is difficult.

Fixed rules cannot be mapped directly on to particular facts or events. Suppose we observe Icabod and Isabel playing a game with chess pieces. They have made five moves and each move has been in accordance with the rules of chess. This may give some evidence that they are playing chess but it does not entail that they are playing chess. On the sixth move Icabod might move his Queen from D1 to E8. This would show that it is not chess, but might be QChess, that they are playing. Even if the game was over and had been played in accordance with the rules of chess this would still not entail that it had been a game of chess. Icabod and Isabel might have been playing QChess and moving in accordance with QChess strategy but neither found it expedient to make one of the peculiar QChess Queen moves. The statement "Icabod and Isabel have been playing chess" is therefore contingent and not logically necessary. It is not, therefore, a straight forward mapping of the logically necessary rules of chess.

5.4 A scientific model

The method that will be used to map the logically necessary model shown in figure 16 onto the real world will be to insert contingent universals, inductive hypotheses, between the necessary universals and the particular statements of real world fact. In this way the required contingency is introduced into the model but the necessary universals still serve a useful function.

The particulars and contingent universals will be expressed in English. This is for ease of exposition. Particular statements and contingent universals could be developed in the gor tonking language but to achieve particular reference would require ostensive definition or an observable language game. That is, to establish a direct reference for the term "gor" we would need to point to one. This cannot be done in a essay, it can only be done in the real world. Thus we shall have to use English terms which already have an established reference.

Figure 16 shows what must be true, as a matter of logic, if the desired state of affairs is to come about. Figure 17 is an attempt to describe a real world state of affairs that will bring it about. This state of affairs may happen to exist. If it does not, the next stage, stage 6 in the learning cycle, is to make changes in the real world that will actually bring it about. The figure 17 model will tell us when and if the desire state of affairs exists.

The figure 17 model consists of particular statements *u*, *v*, *a*, *b*, *d*, *e* and *g* which are putatively true and have real world reference. In addition there are contingent universal statements *t*, *w*, *c* and *f*, these are the inductive hypotheses. In contrast to definitions, they are always open to falsification by particular events. For example, if *u*, *v*, *a*, *b*, *d*, *e* are true and *g* is false then we know that at least one of the inductive hypotheses, t, w, c or f, must be false. We know this because the fact that "q and r and s imply p" is true by definition and so cannot be false. The letter "M" alongside the broken arrows indicates that they are contingent relations. The model can be expressed in modal propositional logic as follows:

$M\ (t\ \&\ u\ \&\ v) \rightarrow q$

$M\ (w\ \&\ a\ \&\ b) \rightarrow r$

$M\ (c\ \&\ d\ \&\ e) \rightarrow s$

$L\ (q\ \&\ r\ \&\ s) \leftrightarrow p$

$M\ p \rightarrow (f\ \&\ g)$

The model would be better expressed in a modal form of the predicate calculus and we have now developed the apparatus for doing this. However, the complications of using this powerful but unfriendly logic will be left to the next chapter where its connection with Prolog will be more easily perceived.

The inductive hypotheses allow contingency into the system and this gives the model the logical flexibility to map on to a contingent world. It is also "scientific" in Popper's sense of the word because the figure 17 model is always capable of being falsified by particular events. This solves the problem of how a conceptual model that is true by definition can map on to a contingent world - the answer is that the mapping itself is contingent.

This has profound consequences for information system design. Information system design methodologies do not distinguish between inductive hypotheses and definitions. What actually happens is that they treat inductive hypotheses as though they were definitions - they make them fixed in the system. When this is done statements of the type t, w, c and f become necessarily true and the broken arrows in figure 17 change from having the "M" modal operator to having an "L" modal operator. When this happens the particular statements u, v, a, b, d and e cannot be true when g is false. Given that a computer systems configured in this way accepts that g is false it will refuse to accept u, v, a, b, d and e as true, even when they are in fact true. In other words it will only map on to the real world when it wants to; but in this case it does not really map on to the real world at all. The only reason that such computer systems can ever work in a changing world is because the human operators apply informal mapping techniques.

It might be thought that it is the inductive hypotheses that are the important thing and that the definitions can be dispensed with, but this is not so. The argument that definitions are logically required in the formulation of any inductive hypothesis was given in section 4.5.1. The usefulness of the definition in the gor tonking language, "$L(q \& r \& s) \leftrightarrow p$", is demonstrated because certain deductions cannot be made without it. For example we can deduce g from f & t & u & v & w & a & b & c & d & e; but we could not do this without "(q & r & s) <-> p".

6 KNOWLEDGE ELICITATION & REPRESENTATION

6.1 The thesis as a language game

The solution to the real world mapping problem given in chapter 5 can now be use as a principle in the design of a knowledge based system. The logical apparatus that has been built up in response to theoretical requirement in chapters 2 through 5 will be used as a practical method. The previous chapters have been largely concerned with showing what is logically required in a method of information system design. It is contended that if any procedure for information system design is to be successful these requirement must either be fulfilled before the procedure takes place or must be fulfilled, tacitly or overtly, by the procedure. The method that will be put forward in the following sections aimed to make explicit what must be assumed or implicit in any method of information system design.

This might sound as though the method is being put forward as the definitive method of information system design. It needed to be emphasized that the method is just *one way* of making the implicit explicit. The language game theory applies to this writing of this thesis just as much as it does to any other form of language based activity. The thesis is part of a language game that is played by philosophers of the Anglo-American analytical school. This is the school that forms the main stream of philosophy teaching in British universities. A good deal of the foregoing chapters have been devoted to explaining how the game is played. Hopefully some computer scientist and some members of business studies departments will be able to understand this game. Even more hopefully some of them might decide to start playing the game.

There might or might not be one definitive method of information system design. If there were such a method there would not be just one way of describing it. The case is well illustrated by Checkland's writing on information. Checkland is not an analytical philosopher or a logician nor, for the most part, are his readers. Checkland is playing a different language game. Thus the word "valid" has a different meaning in Checkland's game from the meaning it has in the game of analytical philosophy. Thus in section 3.4.3 there was no attempt to say that Checkland has got it wrong because his usage of the word "valid" does not conform to the usage in the logic textbooks. Instead the section attempted to translate the sentiment from one language game into another. It attempted to show what Checkland would have said if he had been playing the analytical philosophers' language game.

The thesis is written in the language of analytical philosophy and in modal predicate logic because the author believes that these are the most powerful tools to address the problem.

6.2 Modular structure of the method

This chapter will attempt to put forward suggestions for practical methods and to give the theoretical justification for them. It consists of six main sections. These are concerned with systems analysis, language creation, knowledge elicitation, knowledge representation, codification and verification. Each section begins with a model, then the shortcomings of the model are explained and a remedy suggested, leading to the model in the following section. The method has a modular structure. Each model has a number of uses and need not be merely a stepping stone to the next model. In practice some stages might be unnecessary or they might be achieved by other methods.

In stage one SSM is used for systems analysis. In stage two the consensus SSM model is then developed into a logico-linguistic model by the continuation of the stake-holders iterative debate. In stage three the logico-linguistic model gives a logically precise artificial language that provides an essential framework for knowledge elicitation and the construction of an empirical model. In stage four the empirical model is expressed in the formulas of a modal predicate logic in which formal inferences can be made and which can provide a formal specification for a program for a knowledge based system. In stage five a Prolog program is written based on the modal predicate logic. This program is capable of distinguishing between definitions, i.e. rules taken from the artificial language, and inductive hypotheses, i.e. empirical rules take from the knowledge elicitation process. This enables the program to accept data entry in the form of particular facts that conflict with its empirical rules. In stage six empirical rules that have been falsified by particular facts are recognized as data is entered into the system.

6.3 The six stage method

6.3.1 Systems analysis

Provided "systems analysis" is understood in a suitable broad sense, SSM is primarily a methodology for systems analysis. It claims to be relevant to any problem situation involving human activity. The early stages of the method are more concerned with the identification of who is involved in the problem and what the problem is than with the solution of the problem. SSM is not an information system design method as such but a general problem solving method which may be used in the production of an information system design as one of many possible solutions.

SSM therefore has a versatility not found in main stream information system design methodologies. The use of an information system design method should be based on some form of system analysis that indicates that an information system will be a solution to the problem. In the case where the system analysis is a front end part of an information system design methodology the work of systems analysis will tend to be wasted if an information system is not required. In the case where the method of system analysis is distinct from the information system design method they will tend to use different tools and have different perspectives, as a result little of the information gained in system analysis will be used in the design process.

There is another possibility - one in which there is a continuity between the general systems analysis of SSM and the design process. The idea of preserving this continuity is implicit in Wilson's work and in Multiview. But while Wilson and Multiview add to the stake-holder constructed conceptual models in order to create information system, the method described below seeks to increase the logical power and content of these models to the point where an information system design can be derived by formal methods. The next stage is, therefore, to develop the SSM models into a logically precise language.

6.3.2 Language creation

The vocabulary of the new language will be provided entirely by the stake-holders. We shall take a modal predicate logic as the syntax of the language. This will require an increase in the number of logical connectives used in the model building process. The model will need to have logical connectives, described in chapter 2, capable of expressing causal sequences. There are two reasons. One is that these connectives will be needed later when we come to build the empirical model. The other is that they are needed immediately to express process definitions. The result will be a logico-linguistic model. Figure 19 shows a logico-linguistic model that has been built out of the connection between "Apply treatment" and "Discharge patient" in the Wilson model shown figure 11.

Chapter 5 introduced an "AND" containing box representing conjunction (p and q). Figure 19 and 20 introduce two new types of containing box. These are an "ANDOR" containing box representing inclusive disjunction (p or q or both) and an "OR" containing box representing exclusive disjunction (p or q but not both). "AND" is denoted by the logical symbol "&" "ANDOR" by "v". The "OR" connective can be expressed as "$(p \,\&\, -q) \vee (-p \,\&\, q)$. Figure 19 can be expressed in the propositional calculus as follows:

$$((s \,\&\, a \,\&\, b) \leftrightarrow t) \,\&\, (s \leftrightarrow (u \vee y \vee w))$$

This can be rendered in English as "Patients are discharged if and only if they are alive, have been signed out and have been treated; and Patients will have been treated if and only if they have had surgery, medicine or therapy".

6.3.3 Knowledge elicitation

The logico-linguistic model provides a framework that will enable us to build an empirical model without ambiguity. In the empirical model putative facts about the real world will be added to the logico-linguistic model.

It was established in chapter 4 that there are two types of definition: connotative or intensive definition and denotative or extensive definition. An intensive definition will give a criterion or criteria for class inclusion. An extensive definition will specify the members of the class. Thus in figure 19 "patient is discharged" is given an intensive definition. What it says is that anything that fulfills the criterion of being a treated patient, a living patient, and a signed out patient is a member of the class of discharged patients, and vice versa. "Patient is treated" is given an extensive definition. The figure states that this class has three member classes (*u*, *y* and *w*) and only three member classes. Therefore, anything that is a member of one or more of the member classes will be a member of the class of patients treated, and anything that is a member of the class of patients treated must be a member of at least one of the member classes.

It is contended that any term that is given a useful intensive definition will have an empirical extension, and any term that is given a useful extensive definition will have an empirical intension. Knowledge acquisition, therefore, is simply finding the intension for an extensive definition and the extension for an intensive definition. Knowledge elicitation will amount to the process of the specification of the intension for the extensive definitions and the extension of the intensive definitions by the stake-holders.

The intension of "Patient is treated" might be that every patient has been attended to by a doctor or nurse who has taken some action that is believed to improve the patient's health. Empirical intensions are not always particularly useful. Far more important are the empirical extensions of intensive definitions. In figure 20 we will take the extension of "Patient is discharged" to be the class that comprises the class of patients who return home and the class of patients who are transferred to other institutions. These classes are mutually exclusive in that a member of one cannot be a member of another, as such they are included in an "OR" bubble. This extension is putatively true as a matter of empirical fact not as a matter of definition. It is, therefore, marked with the "*M*" modal operator. These empirical counterparts of definitions are the inductive hypotheses.

The fact that bubbles *c* and *d* are linked to bubble t by a double headed arrow indicated that we think that the formula "$c \vee d$" forms the full extension of *t*. In this case we have full knowledge of *t*. In practice the knowledge of the stake-holders may be insufficient to give the full empirical extension for every intensive definition in the system. In this case there three possible courses of action. One is to conduct empirical research in order to find the full extension. A second is to build a system with incomplete knowledge; if this is done the system will be logically incomplete and there will be statements that are undecidable - that is the system will not be able to determine whether they are true or false. A third possibility is for the stake-holders to make an educated guess and hope that the system will detect any errors. A system of non-monotonic logic is introduced below which makes this third possibility viable.

6.3.4 Knowledge representation

In this section figure 20 will be expressed in predicate logic. What the model actually states is expressed in three premises. From these premises formal inferences can be made which leads to a conclusion (this is (15) below) that is not immediately obvious. This conclusions is added to figure 20 to produce figure 21. Although readers might be able to infer this conclusion intuitively, in larger models there would be inferences that were far from being intuitively obvious.

The formal rules of inference and replacement for the predicate inferences that will be used here are fairly standard and conform closely to those found in Copi (1968), Newton-Smith (1990) gives a similar set. To determine the modal operators four meta-rules are used that follow from the axioms of the modal system "S5". The meta-rules will be expressed using the syntactic turnstile which will be indicated by the symbol "¦-". In this system "A ¦- B" means B can be derived from A, or to express this another way B is provable from A. These meta-rules are:

Meta-rule one: if A ¦- B then *L*(A) ¦- *L*(B)

Meta-rule two: if A ¦- B then *M*(A) ¦- *M*(B)

Meta-rule three: if A, B ¦- C then *L*(A), *L*(B) ¦- *L*(C)

Meta-rule four: if A, B ¦- C then *M*(A), *L*(B) ¦- *M*(C)

That these meta-rules apply is intuitively obvious. However, for those interested in formalism a formal proof is provided in Appendix 2. Meta-rule one simply states that if B can be derived from A then if A is logically true then B must be logically true. Meta-rule two states that if B can be derived from A then if A is contingently true then B must be contingently true. Meta-rule three states that if C can be derived from A and B then if A and B are logically true then C must be logically true. Meta-rule four states that if C can be derived from A and B then if A is contingently true and B is logically true then C must be contingently true.

Figure 20 can be formally expressed in modal predicate logic as follows:

Domain: people who go to hospital

Sx: x is a patient who is treated

Ax: x is a patient who is alive

Bx: x is a patient who is signed out

Tx: x is a patient who is discharged

Ux: x is a patient who has surgery

Yx: x is a patient who has medicine

Wx: x is a patient who has therapy

Cx: x is a patient who returns home

Dx: x is a patient who is transferred to another institution

Prem (1) *L* (Ax) Tx <-> (Sx & Ax & Bx)

Prem (2) *L* (Ax) Sx <-> (Ux v Yx v Wx)

Prem (3) *M* (Ax) Tx <-> ((Cx & -Dx) v (-Cx & Dx))

(4) *L* (Ax) (Tx -> (Sx & Ax & Bx)) & ((Sx & Ax & Bx) -> Tx)

From (1) by Material Equivalence and Meta-rule one

(5) *L* (Ax) (Sx -> (Ux v Yx v Wx)) & ((Ux v Yx v Wx) -> Sx)

From (2) by Material Equivalence and Meta-rule one

(6) *M* (Ax) (Tx -> ((Cx & -Dx) v (-Cx & Dx))) & (((Cx & -Dx) v (-Cx & Dx)) -> Tx)

From (3) by Material Equivalence and Meta-rule two

(7) *L* (Ax) Tx -> (Sx & Ax & Bx)

From (4) by Simplification and Meta-rule one

(8) *L* (Ax) -Tx v (Sx & Ax & Bx)

From (7) by Material Implication and Meta-rule one

(9) *L* (Ax) (-Tx v Sx) & (-Tx v Ax) & (-Tx v Bx)

From (8) by Distribution and Meta-rule one

(10) *L* (Ax) -Tx v Sx

From (9) by Simplification and Meta-rule one

(11) *L* (Ax) Tx -> Sx

From (10) by Material Implication and Meta-rule one

(12) *L* (Ax) Sx -> (Ux v Yx v Wx)

From (5) by Simplification and Meta-rule one

(13) *M* (Ax) ((Cx & -Dx) v (-Cx & Dx)) -> Tx

From (6) by Simplification and Meta-rule two

(14) *L* (Ax) Tx -> (Ux v Yx v Wx)

From (11) and (12) by Hypothetical Syllogism and Meta-rule three

(15) M (Ax) ((Cx & -Dx) v (-Cx & Dx)) -> (Ux v Yx v Wx)
From (13) and (14) by Hypothetical Syllogism and Meta-rule four

Premise (1) can be expressed in English as "For all x, x is a patient who is discharged if and only if x is a patient who is treated and alive and signed out. Formulas which being with (Ax) are known as "universals" as are the English statements that correspond to them. The formula (15) can be deduced from the three premises, it is shown in figure 21 by the dotted arrow and the solid single headed arrow. This completes the system of universals, but so far it is only about object variables, in this case "x". It says nothing about the real world, not even that anything exists. The real world connection is made when particulars and existential statements are added to the system. We shall not introduce particulars into this system of predicate logic instead we shall move on to Prolog where the universals shall become "rules" and particulars will become Prolog "facts".

6.3.5 Codification

6.3.5.1 The Prolog model

The horn clauses that form the logical format of all Prolog rules could be derived from the predicate logic given above. As a practical method this would not be very useful. For example, the formal derivation of (15) from the premises given is very lengthy. It is easier to look at figure 21 when writing the Prolog or to look up the relevant Prolog rule in the dictionary given at 11.3.2. Also it is open to question whether all logico-linguistic models can be expressed in horn clauses. The formal logic is useful in a number of respects as will be seen in later sections. However, a full discussion of the relationship between predicate logic and Prolog is beyond the scope of this thesis.

The program given here is written in Turbo Prolog. There are two serious difficulties in converting the logic, or a logical model like figure 21, into Prolog: one is with negation the other is with the biconditional. Turbo Prolog will not compile rules which begin with the "not" predicate. Therefore, there is no straight forward way to express horn clauses with a negative consequent. This is a problem because some formulas in predicate logic cannot be expressed in horn clauses with positive consequents. The biconditionals express mutual implication as is indicated by the double headed arrows in figure 21. As it works on the chaining principle Prolog is unable to run a program that contains mutual implication, or any substitute for it, without going into an infinite loop. For example, if we want to express the biconditional between *t* and *b* & *a* & *s* from figure 21 we would expect to be able to express it in a logically equivalent form such as (*t* -> (*b* & *a* & *s*)) & (*b* -> *t*) & (*a* -> *t*) & (*s* -> *t*) this is done alone with instantiations in Program 1.

Program 1

```
discharge (X) if sign_out (X) and alive (X) and treated (X).
discharge (icabod).
sign_out (X) if discharge (X).
sign_out (isabel).
alive (X) if discharge (X).
alive (isabel).
treated (X) if discharge (X).
treated (isabel).
```

Although Program 1 will compile it will not run. Prolog looks for a value for discharge (X) and sees that it will have the same value as sign_out (X) and alive (X) and treated (X); it then looks for a value for sign_out (X); on the third line it sees that sign_out (X) has the same value as discharge (X); returning to the first line it tries to repeat the process infinitely.

The solution is to replace straight forward negation, which is troublesome anyway, with a substitute program in the Prolog programs. This can be achieved in the same way in which subtraction is eliminated from commercial accounts by a system of double entry book-keeping. We shall use artificial predicates prefixed by "not_" to express negation. Corresponding to these will be artificial objects also prefixed by "not_". A positive predicate will always be paired with a negative predicate and a positive object paired with a negative one. Thus if we which to say Isabel is alive we will also say that not-Isabel is not alive:

```
alive (isabel).
not_alive (not_isabel).
```

The two negatives can be understood as cancelling each other out. We can also use this method to specify events that have not happened. For example, if Icabod has not had surgery we can say:

```
surgery (not_icabod)
not_surgery (icabod).
```

Program 2 can be put together on the basis of the same particular facts as Program 1, no additional data is required.

```
discharge (icabod).
not_discharge (X) if not_sign_out (X).
```

```
not_discharge (X) if not_alive (X).

not_discharge (X) if not_treated (X).

not_discharge (not_icabod).

sign_out (X) if discharge (X).

sign_out (isabel).

alive (X) if discharge (X).

alive (isabel).

treated (X) if discharge (X).

treated (isabel).

not_sign_out (not_isabel).

not_alive (not_isabel).

not_treated (not_isabel).
```

In Program 2 one half of the biconditional is expressed in "not_" predicates the other half in normal predicates. This solves the infinite loop problem and the program will run. The program will return the same information as would Program 1, if Program 1 could run, but sometimes twice the number of queries are required. For example the query "Goal: discharge (X)" returns only "icabod". We can find out that Isabel has also been discharged by "Goal: not_discharge (X)" which returns "not_isabel" and "not_icabod". This can be read as "it is not the case that Isabel has not been discharged" or simply "Isabel has been discharged". Other queries function as normal, "Goal: treated (X)" returns "icabod" and "isabel". Prolog can work out from its rules that icabod has been discharged even though this has not been specifically stated. The program therefore confers all the advantages of a Prolog style program over an SQL and data base system.

Some Prolog programmers might consider that synonyms should be eliminated prior to writing the program. This would certainly save space but increases the amount of logical work that the programmer needs to do. Ultimately even more space could be saved by not writing the program at all; this is because there is nothing that Prolog can work out that could not be worked out manually using the predicate calculus. A logic program that does not do much logic is not much of a logic program.

These double entry procedures will enable us to express the three biconditionals from figure 20 in Prolog. However, they do not produce concise programs. Indeed the Prolog given at 6.3.5.2 represents what can be expressed in four lines of predicate logic e.g. the three premises and the conclusion (15) from section 6.3.4.

A second difficulty is that although the double entry proceedure solves the immediate problem with the biconditional, it does not solve all the problems of expressing predicate logic in Prolog (see section 10.3.4).

The last four lines of the Prolog are concerned with verification.

6.3.5.2 A Prolog program

Clauses

```
surgery (not_jack).
surgery (X) if not_medicine (X) and not_therapy (X) and
returns_home (X).
surgery (X) if not_medicine (X) and not_therapy (X) and
another_institution (X).
medicine (not_jill).
medicine (X) if not_surgery (X) and not_therapy (X) and
returns_home (X).
medicine (X) if not_surgery (X) and not_therapy (X) and
another_institution (X).
therapy (not_jill).
therapy (not_jack).
therapy (X) if not_surgery (X) and not_medicine (X) and
returns_home (X).
therapy (X) if not_surgery (X) and not_medicine (X) and
another_institution (X).
returns_home (jill).
returns_home (not_jack).
another_institution (not_jill).
another_institution (jack).
not_surgery (jack).
not_medicine (jill).
not_medicine (jack).
not_therapy (jill).
not_therapy (jack).
not_returns_home (jack).
```

```
not_returns_home (not_jill).
not_another_institution (not_jack).
incorrect_hypothesis
(surgery_if_not_medicine_not_therapy_and_returns_home)
if not_surgery (X) and not_medicine (X) and
not_therapy (X) and returns_home (X).
incorrect_hypothesis
(surgery_if_not_medicine_not_therapy_and_another_institution)
if not_surgery (X) and not_medicine (X) and
not_therapy (X) and another_institution (X).
incorrect_hypothesis
(medicine_if_not_surgery_not_therapy_and_returns_home)
if not_medicine (X) and not_surgery (X) and
not_therapy (X) and returns_home (X).
incorrect_hypothesis
(medicine_if_not_surgery_not_therapy_and_another_institution)
if not_medicine (X) and not_surgery (X) and
not_therapy (X) and another_institution (X).
incorrect_hypothesis
(therapy_if_not_surgery_not_medicine_and_returns_home)
if not_therapy (X) and not_surgery (X) and
not_medicine (X) and returns_home (X).
incorrect_hypothesis
(therapy_if_not_surgery_not_medicine_and_another_institution)
if not_therapy (X) and not_surgery (X) and
not_medicine (X) and another_institution (X).
```

6.3.6 Verification

Validation of the program is not a theoretical problem in this system because the rules can be formally derived from the predicate calculus. Any error will be the result of either mistakes made during the construction of the empirical model or mistakes made in entering particular facts into the program. Errors in both respects can be picked up by the double entry system. For example our program in the box produces:

```
Goal: surgery (X)
X = not_jack
X = jill
X = jack
```

This says that Jack *has and has not had* surgery. This could have been a result of a mistake at a data entry level but in this case it is not. The mistake is in the empirical model. The last three lines of the program are designed to detect these errors. The incorrect_hypothesis predicate picks up inductive hypotheses that have been falsified by particular facts:

```
Goal: incorrect_hypothesis (X)

X =
surgery_if_not_medicine_not_therapy_and_another_institution
```

X =

medicine_if_not_surgery_not_therapy_and_another_institution

X =

therapy_if_not_surgery_not_medicine_and_another_institution

Jack has not had surgery, medicine or therapy; he has not returned home but he has been transferred to another institution. The formula:

(15) *M* (Ax) ((Cx & -Dx) v (-Cx & Dx)) -> (Ux v Yx v Wx)

which represents the broken arrow and the single headed arrow in figure 3, is therefore, incorrect. It follows from this that one of the three premises (1), (2) or (3) must be incorrect. As premises (1) and (2) are logically true it must be premise (3), the one with the "*M*" modal operator, that is false. In simple terms the hypothesis that all patients who return home or are transferred to an other institution are discharged patients, has been falsified by a particular event. This event is Jack being transferred to another institution without having surgery, medicine or therapy. The Prolog program has been configured in such a way that the entry of data about Jack has enables us to detect this. This is a form of non-monotonic logic; the program has learned that one of its premises is false.

The benefits of the earlier SSM work can now be seen. The modal distinctions were made using SSM and without the modal distinctions we would not be able to determine which of the three biconditionals in figure 3 is false. Without the modal distinctions all three biconditionals would have the same status. If they all had the status of inductive hypotheses then the fact that Jack been has transferred to another institution without having surgery, medicine or therapy could be equally well explained by "all discharged patients are treated, alive and signed out" being false or by "all treated patients have surgery, medicine or therapy" being false. If they all had the status of logical truth the situation would be even more unsatisfactory.

Consider what would happen if the three biconditionals had the status of logical truth. If this were the case the system would only accept those empirical particulars that are consistent with its in-built logical configuration. All other particulars would be rejected. Consider the biconditional between the t bubble and the bubble containing "c or d", which is expresses as (Ax) Tx <-> ((Cx & -Dx) v (-Cx & Dx)) in predicate logic. If this were a logical truth, then before we could establish that Jack has been transferred to another institution we would have to establish that he had not returned home and that he has been discharged. To establish that he has been discharged we would have to establish that he has had treatment and to do this we would have to establish that he has had surgery, medicine or therapy. In other words to establish that Jack has been transferred we must first establish that Jack has had surgery, medicine or therapy. We need to do this because having surgery, medicine or therapy is part of the extended definition of a patient who has been transferred. But in this case the model does not enable us to infer anything new about Jack at all. All that it says is that if Jack fulfils all the defining criteria then each defining criterion will be true of Jack.

If $(Ax)\ Tx \leftrightarrow ((Cx \,\&\, -Dx) \vee (-Cx \,\&\, Dx))$ is contingent, as it is in the figure, then we can establish that Jack has been transferred by a defining criterion that is independent of the system shown in figures 20 and 21. In this case the system can be genuinely informative and tell us some real world facts about Jack.

Verification unlike validation is only possible if parts of the system is open to falsification. The hypotheses in the system will be verified with the addition of each particular fact that does not falsify them. Systems that can not be verified, even if they can be validated, cannot in themselves refer to real world objects and events. Real world event are contingent and therefore any statement about the real world must also be contingent. As was contended in chapter 5 systems that do not contain contingent elements will only map on to these real world events that they want to and as such do not really map onto the real world at all. Inductive hypotheses form an indispensable buffer between definitions and real world particular facts.

6.3.7 Automatic deletion

In their book on Prolog Clocksin & Mellish (1987, p. 116) describe a "built in" predicate called "retract" which is not present in Turbo Prolog. It is not clear which versions of Prolog contain this "retract" predicate because it not clear which version of Prolog they are describing.

> In this book we have presented a version of Prolog that does not correspond exactly to any existing system. Rather, it is supposed to represent a "core" Prolog that will have a lot in common with any system you might encounter. (Clocksin & Mellish, 1987, p. 260)

The resources available during the research for this thesis were limited to Turbo Prolog where the "retract" predicate is restricted to removing facts. However, it is easy to imagine what a program would look like if it contained the "retract" predicate that could also remove rules. The "retract" predicate functions as follows: in "retract (X)", X will be matched with the first clause (fact or rule) that it can be matched with and that clause will be removed. So it would seem that:

```
retract (surgery (X) if not_medicine (X) and not_therapy (X)
and
returns_home (X)) if not_surgery (X) and not_medicine (X)
and
not_therapy (X) and returns_home (X).
```

would delete the rule:

surgery (X) if not_medicine (X) and not_therapy (X) and returns_home (X).

given an instantiation for:

not_surgery (X) and not_medicine (X) and not_therapy (X) and returns_home (X).

Therefore, instead of an incorrect inductive hypothesis merely being identified, it could be removed by a Prolog program. This would then be a program that embodied Popperian falsification. Inductive hypotheses could removed by particular facts that falsified them.

however, the practical use of a Prolog program that contained this automatic deletion function would be limited to applications where mistakes would not occur in data entry and where the data is reliable. In a normal business system keyboard mistakes would result in the rapid deletion of perfectly good rules. However, in the days of punched cards methods, such as check digits, were developed to prevent this sort of error.

Mistaken beliefs about particular facts would also cause the deletion of perfectly good rules. For example, if it was mistakenly believed that Jack had not returned home when in fact he had, then the perfectly good rule "surgery (X) if not_medicine (X) and not_therapy (X) and returns_home (X)" would be deleted.

6.4 Other types of knowledge representation

A comparison of the foregoing with previous work on knowledge elicitation and representation is difficult to accomplish due to the fact that the dozens of different methods differ in many significant respects. Sowa's conceptual graphs have a bubble diagram style, are concerned with concepts and can be expressed in the predicate calculus; they are, therefore, superficially very similar to logico-linguistic models. Instead of trying to consider all the different forms of knowledge representation the following comparison will be limited to Sowa's conceptual graphs.

Like many other semantic net style graphs the logical and epistemological status of Sowa's graphs is not perfectly clear. He recently described them as "a graphic system of logic ... equivalent to predicate logic" (Sowa, 1992), yet earlier he described them as "a method of representing mental models" (Sowa, 1984, p. 4).

As a graphic system of logic Sowa's graphs differ from logico-linguistic models firstly, in that they do not include modality which is one of the principle features of logico-linguistic models. The second difference is that Sowa's graphs contain the plethora of detail needed to capture the vagaries of English syntax. For example, the verb "to run" is represented by ten bubbles and ten arrows in a conceptual graph (see Nogier & Zock, 1992). Such detail is not essential for the construction of a knowledge based system nor is it practical as a stake-holder driven modelling device.

It is clear that conceptual graphs are a tool for an analyst intending to represent discourse in a natural language. Logico-linguistic models, by contrast, are not intended to represent a natural language but are intended to be an artificial language. Logico-linguistic models are not, therefore, dependent on lexicographical science nor are they prone to the paradoxes of self reference which are a feature of natural languages. It is pertinent to point out here that it may, as both Frege and Tarski believed, be impossible to formulate a theory of truth for natural languages (Grayling, 1990).

It must be made clear here that there is a difference between a computer program that is based on a language and a program that deals with a language. Logico-linguistic models are an artificial device intended to form the basis of a computer language; they are not intended to represent an existing language. In order to represent and deal with an existing language it is not necessary to base the computer program upon that language. For example, we can give a set of instruction in English about how to answer French questions in French; here our program equivalent is English and the coherence of our instructions will be dependent on the coherence of English not French. Sowa's graph might be useful as a description of a natural language that could form the subject matter of a computer language that would deal with queries in the natural language. Unfortunately the language game theory shows that there is no absolute natural language just language games played by individual players. Of course conceptual graphs could be used to represent an existing language game but in this case they would be the same sort of thing as the SAMPO method developed by Auramaki (1988). As was pointed out in chapter 3 this is not suitable for green field situations as is the SSM/logico-linguistic modelling technique.

If Sowa's conceptual graphs are intended to be mental models then it seems they are very different from logico-linguistic models. The Wittgensteinian theory of language contends that mental models, if they are anything other publicly observable neurological states or elements of a public language, simply do not exist. Sowa's discussion of "percepts" (Sowa, 1984, p. 24) sounds very similar the sense datum theories that were discredited by Wittgenstein's private language argument. Logico-linguistic models are not intended to be representations of mental models nor are they representations of anything, they are just records of an agreement to uses words in a certain way.

It seems that the theory of meaning which forms the basis of Sowa's graphs is fundamentally different from the one that is assumed here. This could explain the fact that Sowa's graphs lack the modal operators that form vital components of empirical models such as figure 21.

6.5 The attribution of meaning

Checkland & Scholes claim that an information system:

> "...will always have to include the attribution of meaning, which is a uniquely human act. An information system, in the exact sense of the phrase, will consist of both data manipulation, which machines can do, and the transformation of data into information by the attribution of meaning." (Checkland & Scholes, 1990)

Traditional methods of information system design, such as SSADM and Information Engineering, produce systems in which tacitly all the universals are incorrigible. This prevents them from making statements that refer to the real world. It is up to the human operators to determine reference. If the operators do this, they will do so by formulating inductive hypotheses as well as by observing particular facts.

This paper has provided a description of the logic that would be required for the creation of a computer system that will be able to make statements in a language belonging clients and users. While particular facts need to be entered into this computer system by human beings, the system is able to learn and this one can argue requires some attribution of meaning. In Program 3 we could say that the system knows that the hypothesis "all discharged patients are patients that have returned home or been transferred to another institution" is false because it knows that "patient is discharged" means "patient is treated, alive and signed out". Effectively Program 3 is able to distinguish between sense and reference a thing that systems designed by traditional methods cannot do.

Checkland & Scholes contention that the attribution of meaning is a uniquely human act might be true in the context of systems design by traditional methods but it less plausible in the context of systems such as Program 3.

6.6 The value of the Prolog program

The fact that a Prolog program with automatic deletion of falsified inductive hypotheses can delete perfectly good rules if a mistake is made about a particular fact must be weighed against a conventional system. In a conventional system a particular fact cannot enter the system if it conflicts with one of the rules. Both have their demerits. Suppose that *all crows are black* is a rule of the system. The automatic delete system will delete the perfect good rule that all crows are black if the person entering data has mistaken a white pigeon for a crow. The conventional system will be unable to process the data if the person has in fact observed a white crow.

The respective merits of the two systems depend on which is more important in the context - rules or particular facts. In conventional business systems, such as order processing, rules tend to be more important than particular facts. It is better to reject an order without an order number than to delete the rule that all valid orders have order numbers.

However, in safety critical systems, such as intensive care, the opposite can apply. It is better to falsify the rule *patient is dead if heart has stopped for ten minutes* when the patient is in fact dead but the doctor mistakenly thinks the patient still alive, than it is for the system to refuse to process data about a patient that has recovered from a ten minute heart failure.

Another possible application is in the "find the murderer" problems that have long been a stock example of the power of Prolog. In Warwick Business School students on the operational research course are required to write a program to play Cluedo. Cluedo is a board game in which there are a number of suspects (Colonel Mustard, Miss Scarlet, etc.) for a murder. As the game proceeds clues are revealed and ultimately the winner can work out which of the suspect is the murderer by a process of elimination. It is easy to see how the methods proposed in this chapter could be used to write a program for this game. The rules of the game would form the definitional rules of the program, the inductive hypotheses would take the form "Colonel Mustard is a suspect", the particular facts would be the clues. As more and more clues are entered into the system more and more of the inductive hypotheses would be delete. In the end there would only be one left and that suspect would be the murderer.

Of course, a Prolog program for Cluedo can be written, and no doubt many have, that are more economical than the one suggested here. Also the knowledge elicitation methods suggested here are not required for a writing a Cluedo program. However, the point here is that the definitional rules of the program are fixed by the rules of Cluedo. In the real world there are no self evident definitional rules. To find a real world murderer one would have to determine what to count as fixed rules and what to count as hypotheses and for this one would need something of the same logical order as the method of knowledge elicitation described in this chapter.

There is no doubt that there are more powerful learning programs than that presented in this chapter. Methods of decision tree induction such as Quinlan's ID3 (Forsyth, 1990, p. 202) are able to generate and well as delete inductive hypotheses (Forsyth, 1990, p. 202). However, this is all ID3 does, it cannot handle the database queries and knowledge based system solutions that the program in 6.3.5.2 can.

Therefore, there may be some application for a Prolog program of this type. There would be wider applications if the program is limited to the incorrect hypothesis function and excludes the retract function. Against this it has been argued that other systems written in, for example PASCAL, can do much the same thing. If a program in another language does do the same thing then there will be an implicit distinction between definitions and inductive hypotheses in the system. Given this the language in which the system is written will be capable of representing modality and can be substituted for Prolog at stage five of the method.

It should be strongly emphasized at this point that Prolog is used in this thesis only as an example to show that it is possible to represent the knowledge drawn from the knowledge elicitation method in a computer system. Prolog is used because its near English format makes it easy to read and its proximity to predicate logic make it easy to see how it can be derived from modal logic.

The main problem to be overcome was the real world mapping problem not to achieve a practical artificial intelligence solution to practical problems. If these can be obtained from the use of Prolog programs like 6.3.5.2 then this is an added bonus.

7 LOGICO-LINGUISTIC MODELLING IN PRACTICE

7.1 Summary of the study

7.1.1 Theory and practice

The work in the previous chapters has been driven by theory and very abstract theory at that. This has been in accordance with a methodology of philosophy. The aim has been to show that information systems should be designed in a certain way because, given a set of assumptions, it can be proven as a matter of logical necessity that they must be designed in that way. Empirical research has little or no bearing upon these arguments. However, the reason the philosophical analysis has been undertaken is because it is believed that this will lead to a practical method for information system design. It is at this point that empirical research comes into play.

If "information systems" is widely defined they will effect almost all human activities and we will find a large number of human activities that are solely concerned with information systems. All languages could be described as information systems. If "information system design" is given a similarly wide definition then, again, this will encompass a wide range of human activity. A family deciding what name to give a new pet, the election of a new secretary for the cricket club, a student deciding how to structure an essay and a journalist reporting a wedding could all be described as exercises in information system design. We can call this "informal information system design".

Against this we have "information system design" as in information systems engineering. This is an area for the professional - a person who has passed examinations in SSADM, or some other methodology, and goes through a set procedure that always results in a computer system. We can call this "formal information system design".

The aim of this thesis is to identify suitable principles for formal information system design and one of the requirements of a formal information system design method is that it must be capable of being implemented in practice. So far the thesis has put together a formal method indicated by philosophical analysis. It has been contended that something analogous to the formal method must be implicit in successful practice. What has yet to be demonstrated is that the formal method can be put into practice *as a formal method*. It might be that these methods can only be accomplished in informal information system design. That is, although the philosophical analysis has shown the procedures that must take place it does not follow that these procedures can take place as part of a formal practical method; it might be that they need to be left as informal procedures.

A second point is that a number of devices might be capable of fulfilling the same logical requirement. Some of these may be practical and some might not. Empirical research is, therefore, necessary to determine whether the finding of the previous chapters can be used in a practical context. Empirical research is also necessary in the development of the proposed method from a theoretical framework into a tool honed for practical requirements.

However, this sort of research is not like experimental science. A single case study will neither prove nor falsify the method. Requirements for information systems are so varied that the fact that it works well in one case provides little inductive evidence that it will work in others. Conversely the fact that it fails in one case will provide little inductive evidence that it will fail in all.

This chapter gives an account of how logico-linguistic modelling was used in practice. The project ended with the construction of an empirical model. Sections 7.5 and 7.6 show the techniques that could have been used to convert the model into a relational database or into Prolog. This, however, was not part of the project.

7.1.2 The context

In early 1993 logico-linguistic modelling techniques were used for problem structuring in the context of a public relations firm. The company was located in an Asian country. There was no funding available for the project and the analyst/facilitator, who is also the author of this thesis, was constrained by a limited amount of time in the country. The duration of the project was less than five weeks in elapsed time. The model building took place during a series of discussions with the Chairman of the company.

The analyst was, naturally, looking for an opportunity to apply logico-linguistic models in the context of knowledge based system design. In the event a knowledge based system was inappropriate due to the small size of the project and the limited number of people involved. The useful outcome of the project was more general. This, however, can be seen as a vindication of the use of SSM in the context of knowledge based system design. There were useful results for the customer. If a traditional knowledge engineering approach had been used the work would have been wasted.

The main theoretical finding was that logico-linguistic models can be easily understood by a customer who has no previous knowledge of SSM or of logical modelling. A second finding was that the modelling technique is just as suitable for highly abstract human activities as it is for the physical activities that constitute most of the examples that were used to develop the method. The project was an exercise in action research rather than an attempt to prove the validity of the methods advocated in this thesis. It represents a first step in the extension of theoretical validity to practical applicability.

The client requested complete confidentiality and, therefore, neither the identity of the client and the location of the project cannot be revealed. In the following account the term "the Country" will be used to denote the nation where the public relations project was located, the term "the Company" will be used to denote the public relations company for who the logico-linguistic modelling was undertaken, "the Client" will be used to denote the company for who the public relations were undertaken, "the Products" will denote the items that the client was having trouble with.

The project began at the behest of the Chairman who expressed an interest in using soft methods and logico-linguistic models. The company was newly formed and had been in existence for about two years. The routine running of the company was unproblematic. Most of the projects were handled by in-house staff and administered by the Chairman's partner. The company was operating smoothly at this level and the Chairman spent little time on the day to day running of the Company, the difficulty was with the special projects that he administered himself.

There were two types of special project. The first type were contracts that fell outside the capacity and capability of in-house staff. With these projects the Chairman would put together a team of specialists from outside the company and personally direct the project, other companies would also be involved as sub-contractors. The size of these projects were in the medium range - for example, the most recently acquired contract was worth approximately one million US dollars. The second type of special project were for particularly valuable clients, these the Chairman handled personally.

7.1.3 The problem situation

There were a number of on-going projects and all of them were suitable for logico-linguistic modelling. After some preliminary model building of various projects, the most complex project was chosen as being the most appropriate. This was a public relations project for a multinational client involved in the processing and distribution of food and other consumable products. They were concerned because the Country had recently introduced regulations detrimental to the sales of one of their range of products and certain factions in the government were proposing even more detrimental legislation. The Company was hired to undertake public relations that would prevent further legislation against the products and, if possible, help to repeal the regulations that had just been passed.

The remit for the Company was to undertake research into the social and political situation with regard to the Products and to keep the Client appraised of this situation. On the basis of this research the Company would develop a public relations strategy for the Products. The main vehicles for the implementation of the strategy were likely to be arranging for the case against regulations to be presented in the media and the hiring of lobbyists to influence public opinion.

7.1.4 Modus operandi

The model building process took place in a number of long sessions in which only the Chairman and the analyst participated. In the first session the Chairman described the situation and the analyst took notes. In subsequent sessions the analyst would present models based on the notes from the previous session, the models were then discusses and the Chairman would proceeded to talk about areas of the situation not covered in the models. The analyst was in fact merely returning the Chairman's ideas in a structured form. The learning, therefore, derived from the structuring alone. This can be contrasted with most SSM projects where the learning derives from the exchange of stake-holders' views.

A salient point is that the general Checkland style models which started the model building process held little interest for the Chairman. It was only in the construction of the more complex logico-linguistic models that he thought he was getting somewhere. This was probably due to the fact that there was only one Weltanschauung, that of the Chairman, being represented. The analyst's role was more or less confined to representing the Chairman's viewpoint. Thus there was no real debate or language game in this project. Stage two of the six stage process advocated in Chapter 6 did not conform to the normal pattern and the logico-linguistic modelling was a representation of an existing language rather than a dynamic process of language creation.

The modelling was undertaken without any clear idea about what might result. One possibility was that logico-linguistic modelling might help to find out what would be required for the legal environment to be improved. In the event the empirical model could not be completed in sufficient detail because of lack of knowledge. However, areas where the Company did not have sufficient knowledge could be identified and this was a useful result. It indicated areas where the Company needed to conduct empirical research and also identified some potential benefits that would follow from the research.

7.1.5 The outcome for the Company

The most immediate outcome for the Company was a list of bullet points representing areas where research or action was indicated. These were:

1. Identify Minister's close associates and political allegiance.
2. Identify groups in favor of repealing legislation.
3. Bring these groups to the attention of the Minister (by publicity in the press).
4. Create new or stronger groups (by providing facilities).
5. Map Product cultivation and processing areas against the constituencies of Members of Parliament.
6. Conduct a survey to determine whether the general public is for or against more stringent regulations and the reasons for their opinion.
7. Use the survey results for strategy design.
8. Use the survey as a lobbying device if appropriate results are obtained.
9. Determine the position of the dominant local religion with regard to legislation effecting the Product.

Most of these points had already occurred to the Chairman. He had spent considerable time thinking about the problem situation and had also used some Harvard strategy methods, such and strengths and weakness analysis, to structure the problem. However, some ideas were new and an outcome of the model building process.

During the model building and ancillary discussions it became clear that the vested interest of Members of Parliament would tend to be in favor of the Product in those constituencies where a large number of constituents were employed in the cultivation or processing of the Product. Therefore, the Chairman determined to compile a list of MPs representing these areas and to give them special attention.

The lack of knowledge about public opinion was of particular concern. A survey would stand to provide information for the development of project strategy and the results might also be useful as a lobbying device. The Chairman determined to put a forward a proposal for a survey of public opinion at his next meeting with the Client. However, such surveys are expensive and he doubted that it would be approved. As the model is a form of information requirements analysis the Chairman recognized that the model would form an excellent basis for questionnaire design. This is an idea that had not previously occurred to the analyst.

A less tangible outcome was that the Chairman profited from the structured debate that the model building process provided. As the debate comprised only the Chairman and the analyst the change of perspective was less significant that would normally be the case in an SSM project where numerous stake-holders are involved. The limitations of the debate can, however, be taken as a vindication of the usefulness of the greater detail of logico-linguistic modelling compared to traditional SSM models.

7.1.6 Theoretical findings

The fact that the empirical model could not be completed made apparent interesting implications for the use of SSM in information system design. The foregoing chapters have shown the need for an empirical model if the information system is going to support real world activities. The completion of the six stage process of chapter six assumes that the stake-holders will have the real world knowledge necessary to build the system, if they don't then the method cannot proceed beyond stage two. In traditional knowledge based system design this will not be a problem because the people who built the model will be selected as domain experts - thus the term "expert system".

In the modular structure of the six stage method this is not a problem. If the stake-holders have insufficient knowledge then the method will stop in stage three. The stake-holders can then branch off into empirical research and the logico-linguistic model will help to structure this research. If the research is successful then the six stage process can be resumed.

In traditional uses of SSM for information system design this is a very serious problem. Wilson states that the conceptual model should not represent an existing state of affairs, from this model along with some analysis he produces an information system to support the new activities. Wilson assumes that all knowledge that is needed for the construction of the new information system is already available. There is no place in Wilson's method for empirical research outside the organization.

Suppose we have an organization that makes and retails ice-cream. The stake-holders are not happy with this and construct a conceptual model of a desirable system. Lets suppose the desirable system is a system to make and retail pizza. From this model Wilson then goes through the stages of information categories, Maltese cross and data flow diagrams. When these are completed the programmers can put together a computer system to support the activities involved in the manufacture and retail of pizza.

The stake-holders might like pizza and know that there is a good market for it. But why should we assume that they know how to make it? Why should we assume that they even know anything about what is involved in making it. There is nothing in Wilson's method that will enable them to find out. There is not even anything in Wilson's method that will enable them to find out what type of thing they need to find out. So how can an information system be designed to support these activities? The answer is that cannot. Wilson's method can only work when the stake-holders know everything they need to know about the new activities.

7.2 Model Building

7.2.1 Company level models

In order to illustrate the model building process for the Chairman, and to set the project in context, a generic model of a public relations project was constructed. Following discussion and iteration with the Chairman a root definition, CATWOE (in keeping with the image conscious nature of public relations the mnemonic was changed to ACETWO), measures of performance and a two level conceptual model were formulated.

Root Definition:

A system owned by a client and operated by the Chairman and the Company to obtain and complete a public relations contract within the Country's social situation and in which project income exceeds project expenditure.

ACETWO

Actors: Chairman and staff of the Company

Customer:	The Chairman
Environment:	The Country's social scene
Transformation:	Contract available --> Project complete and fees paid
Weltanschuung:	Every company can benefit from good public relations
Owner:	The client company

Measures of Performance

Efficacy (E1):	Project completed
Efficiency (E2):	Income/expenditure
Effectiveness (E3):	High standard of living and enjoyable work for members of the Company

On the basis of this two models were constructed. One (figure 24) was of the ongoing company activities, the other (figure 25) took the "undertake project" element to a higher resolution level. These general level models were successful in acquainting the Chairman with the model building process. The next stage was to select a project for more detailed modelling. This occasioned considerable discussion. There were some half dozen on-going projects but none were of sufficient size or permanence to justify the creating of a computerized information system. Nor was there sufficient time available to build a working system though a demonstrator system was still a possibility. It was, after some preliminary model building of various projects, decided to proceed by modelling the most complex project in the hope that the model building process would, itself, be enlightening. This was the public relations project for the multinational worried about detrimental legislation.

7.2.2 General project models

Figure 26 and Figure 27 give higher level models for the project. Figure 26 is a standard human activity system conceptual model. It is the only one that contains measures of performance. Subsequent models grow out of this model and can be understood as inheriting the measures of performance from it. Figure 27 is element 1 from figure 26 taken to a higher resolution level.

Figure 28 is the beginning of the model itself. It is at a lower order to figures 26 and 27 which are about model building. This is apt to be confusing for people unfamiliar with logical hierarchies. Figures 26 & 27 are at the same level as Checkland's ILSD model (Checkland, 1989b) which is *a system to build an information system* not a model of the information system itself.

A short example might help to make this clear. Suppose we want to build an information system to support pizza retail. We could build a model of a system to build the information system (figure 27 is of this type). This might include activities such as "buy computer" and "model the activities the information system will support". The model of the activities the information system will support will be built within the wider model of the system to build an information system. The model of the activities the information system will support (figure 28 is of this type) will contain activities such as "obtain tomatoes" and "cook pizza" it will not contain activities such as "buy computer". By a series of additions Wilson develops this type of model until it is represented in computer code. In logico-linguistic modelling the stake-holder driven process is extended until it is capable of becoming (by derivation rather than addition) computer code.

The distinction between *system to build an information system* and *information system* starts to overlap when the information system contains non-monotonic logic, self verification and learning ability. When this is the case, as it is in the chapter 6 model, the information system starts to build itself.

The logico-linguistic model shown in figure 28 is a model of the state of affairs in which the criterion for efficacy (E1) is met. In figure 26 this given as "legal environment improved or maintained" this is rendered in element 1 of figure 28 as "Existing regulations repealed and no new regulations introduced". The logico-linguistic model is the model that will show us how to solve the problem while figures 26 & 27 show us how to build the logico-linguistic model.

7.2.3 The logico-linguistic model

Figure 28 gives a pure logico-linguistic model. Figure 29 gives a model that is partially empirical because it comprises an inductive hypothesis, indicated by the "*M*" modal operator, in addition to definitions, indicated by the "*L*" modal operator. A good deal of time was spent constructing figures 28 and 29 the construction was complicated by the fact that it was not easy to determine how laws and rules should be represented in modal logic (see section 7.3). Figure 29 contains what is necessary to build an inductive knowledge based system.

Figures 30, 31, 32, 33 and 34 are not complete in the sense that they do not contain the modal operators needed to construct an inductive system. By the time these model were built it had become evident that there would not be enough time to build even a demonstrator computer system. It was, therefore, decided to use the time available to expand the model in terms of extent rather than in terms of detail.

In figure 30 modal flags are absent because it was never determined whether elements 19, 20, 21 and 22 were an extensive definition of 17 or whether their conjunction represented a hypothesis as to the full extension of 17. If they were a hypothesis then a corresponding intensive definition would need to be added. That is, we might be saying that, as a matter of fact, the only time an official is willing to make a proposal is when it is of direct gain or when it will help his career or when it supports his friends or when he likes the Product. In this case we could be wrong because some other factor might cause him to be willing to make the proposal. And if this is true we need some criterion to determine that he is willing that is independent of the four conditions already specified.

In figures 31, 32, 33 and 34 neither modal operators nor the biconditional are used. They are similar to traditional SSM models and contain either incomplete defining criteria or empirical causal conditions. The work needed to covert these into full definitions with their full extensions was not undertaken.

The model represented in figures 28 through 34 contains 49 elements. More could be easily constructed by following some of the branches which have been ignored for the sake of brevity. For example, figures 29 onwards only develop element 7,"Proposals for repeal are made and passed at Ministerial level". The development of element 6, "Proposals for repeal are made and passed at Parliamentary level" would be very similar to that of element 7. The main difference would be that the word "Parliament" would replace "Ministry" and "Member of Parliament" would replace "Ministry Official". The development of elements 9 and 8 would also follow a similar pattern except that we would be talking about proposals not being passed rather than proposals passed.

In each of these four forks many the motivating factors would end up being the same i.e. personal gain, ethical reasons etc. The model grows exponentially in the first instance and then begins to converge. For example, if it is a fact that the Product has little bad effect on people then this could be part of the cause of Officials or Members of Parliament thinking that a proposal for the repeal of legislation is ethical. It could also be part of the cause of the majority of the public being in favour of repeal which in turn could cause Ministry Officials and Members of Parliament to think that it was in their career interests to propose repeal.

It was this convergence that produced the idea of identifying Members of Parliament with constituencies in Product growing and processing areas. The convergence also indicated that a survey of public opinion stood to be the most valuable piece of research.

7.2.4 Findings from the modelling process

In chapter 6 the logico-linguistic model and the empirical model were presented as distinct stages in the six stage process. This was the most convenient method of exposition as the two types of model are logically distinct. In the project the logico-linguistic model building and the empirical model building were not separated.

The Chairman did not have much difficulty with the notion of modality but there was difficulty in determining whether any given double headed arrow was a definition or an inductive hypothesis. This should not be surprising as people tend to assent to propositions without knowing whether they are definitions or not. Inferences can still be made without this distinction. Failure to make the distinction is just as common among academics as it is among uneducated people. The author is reminded of a discussion at a United Kingdom Systems Society seminar in which the question of whether information is physical was being discussed. The participants engaged in lively debate but it was by no means clear what the debate was about - it could have been about whether "information" should be defined in terms of physical attributes or whether "information" independently defined is in fact physical.

In this instance the analyst felt that it was better to get on with extending the model and to flesh out details later. However, this procedure might not normally be the most suitable. This implementation detail is something that will have to be determined over a number of real world projects.

The rapid growth of the number of elements might be disturbing for someone used to SSM modeling. But the design of a moderate knowledge based system design will require thousands of rules; and data flow diagrams of moderate size operations often need hundreds of pages. It is obviously impractical for stake-holders to spend the same amount of time debating the details of a thousand element empirical model that they would spend on debating the details of a Checkland model. However, it seems unlikely that this much debate would be needed in practice. The most time consuming part of the modelling process is the debate about the relevant system. One this is agreed the time spent on each additional element becomes less and less. It becomes increasing obvious how the model should develop and more and more of the work can be undertaken by the analyst.

One can expect that the role of the analyst will change as the project proceeds. Initially the analyst will mainly be concerned with Checkland style group facilitation. As the model proceeds more time will be spent with individual experts supplemented by desk research. As the model grows it can be submitted to the stake-holders for comment but it can be expected that approval will become increasingly automatic as the project proceeds.

7.3 Law, rationality and modality

The Chairman chose this particular project as the case for study because of all his on-going project this was the most difficult to structure. Another way of saying this is to say that the project was the most difficult to model. The difficulty arose not because of problems with the modelling tools *per se* but because of inherent difficulty in understanding this type of situation. There were two fundamental problems: one was in representing law in terms of modal logic the other was concerned with the problems of modelling belief systems.

Modal distinctions are fairly simple when we are dealing with statements about physical events. With abstract entities the case is more difficult and contentious. The law is a case in point. We can distinguish between a statement *of* the law and a statement *about* the law, for example the following on a sign in Kenilworth:

Littering is forbidden

would be a statement *of* the by-law in Kenilworth. This statement could be taken to be true but it is true as a matter of stipulation rather than as a matter of empirical fact. As such it should be expressed as follows:

Fx: x is an act of littering in Kenilworth
Gx: x is a forbidden act

L (Ax) Fx -> Gx

However, if somebody asserts "Littering is forbidden in Kenilworth" then this is contingent as the person that states it could be mistaken. It should presumably be stated as:

M (Ax) Fx -> Gx

This is not just a problem with modal logic but part of a set of philosophical problems concerned with intentionality. An intensional act, such as a belief, is one directed towards an object, such as a proposition. (Intentionality here is not the used in the same sense intension and extension). A solution is well beyond the scope of this thesis.

The second major problem with the model was that it was largely concerned with what is believed rather than with what is a fact. In the case of rational belief we can undertake some procedure to determine the cause. But this is not possible with irrational belief as literally anything can be the cause of irrational belief. As hardly anybody is rational all the time, and some people are irrational most of the time, this makes for difficulties in the construction of the model. The same difficulty occurs in determining the movement of share prices as these movements are caused by beliefs rather than facts. In economics the normal solution is to use Rational Expectations as the basis for stock market modelling. The same might be appropriate with other belief systems such as public relations.

These problems have a theoretical and a practical aspect. From the theoretical point of view the fact that the logico-linguistic method encounters problems with the law and with belief can be seen as a vindication of the method. This is because the method surfaces problems that other methods are incapable of recognizing. SSADM treats everything as logically necessary. This does not solve the problem it simply ignores it. When systems designed on the basis of SSADM start to go badly wrong, because of a failure to make modal distinctions or because belief systems are represented in the computer system, the designers are quite incapable of understanding the problem.

When a theoretically sound solution is not possible a practical accommodation can still be reached. If the project had proceeded to the creation of a computerized knowledge based system, the analyst would have suggested that the model be built with laws treated as logically true and beliefs modeled on the basis of rational expectations. However, it would have been explained to the stake-holders, with considerable emphasis, that such a system could fall into error if the laws were not correctly represented or if the laws change. The system would also fall into error when the people represented in the system did not behave rationally. Given that the stake-holders knew how the system could go wrong they would be more able to recognize when it did go wrong and be more able to compensate or correct such errors.

The increased number of logical connectives in logico-linguistic modelling, therefore, cause a range of new modeling problems not found in traditional SSM. They raise questions that give philosophers and economists endless trouble and we cannot expect stake-holders or analysts to be able to solve all them.

There are limits to the client driven design here because the client will not know what the model should look like. The analyst will need to be far more active than in traditional SSM and will often need to draw on modelling solutions drawn from other disciplines.

7.4 Converting the model into predicate logic

The bubble diagrams from figure 29 of the case study can now be expressed in the predicate calculus. The numbers will stand for the contents of the bubbles. The logical form of the arrows between the bubbles will be indicated by a slash. Thus 7/10,11 is the arrow between bubble 7 and the box containing bubbles 10 and 11. The contents of the bubbles will always be particular existential statements while the arrows will always be universals. We shall only need three object variables.

In accordance with Ockham's razor (entities must not be increased beyond necessity, see section 7.5.4) we will only have three objects. The Minister could be formulated as a object but there is no need to do so. For any given Ministry there is one and only one Minister.

Hx: x is a proposal for repeal

Ix: x is a proposal passed

Kx: x is a proposal made at Ministerial level

Lx: x is a proposal sanctioned by the Minister

Mx: x is a proposal approved by Cabinet

Rz,x: x is a proposal that is made by Ministry Official z.

Oz,x: z is a Ministry Official in a position to make a proposal x.

Nz,x: z is a Ministry Official willing to make a proposal x.

Pz,x: z is a is a Ministry Official who thinks the proposal x is ethical

Qz,x: z is a Ministry Official who thinks making proposal x will result in personal gain.

Sz,x: z is a Ministry Official who registers proposal x

7: (Ex) Kx & Hx & Ix

10: (Ex) (Ez) Rz,x

11: (Ex) Kx & Ix

12: (Ex) Lx

13: (Ex) Mx

14: (Ex) (Ez) Nz,x

15: (Ex) (Ez) Oz,x

16: (Ex) (Ez) Pz,x

17: (Ex) (Ez) Qz,x

18: (Ex) (Ez) Sz,x

7/10,11: *L* (Ax) (Az) (Kx & Hx & Ix) <-> (Rz,x & (Kx & Ix))

10/14,15: *L* (Ax) (Az) Rz,x <-> (Oz,x & Nz,x)

11/12,13: *L* (Ax) (Kx & Ix) <-> (Lx & Mx)

14/16,17: *M* (Ax) (Az) Nz,x <-> (Pz,x v Qz,x)

Rule 14/16,17: is an inductive hypothesis. So, by the arguments made in chapter 4 it requires an independent defining criterion. This can be established by bubbles 15 and 18.

L (Ax) (Az) (Sz,x & Oz,x) -> Nz,x

This means that, as a matter of logic, it will be true to say that an official is willing to make a proposal if he is in a position to make the proposal and if he registers the proposal. Establishing that an official is in a position to make the proposal and establishing that he has registered it will be established independently of establishing that the official thinks the proposal is ethical or will result in personal gain. We include the statements about ethics and personal gain because we are looking for the cause of an official being willing. The defining criterion is there to check if were right in our causal account.

7.5 Deriving a relational database

7.5.1 Entity-relationship modelling

The fundamental principle behind Relational Database design is to maximize the efficiency of computer storage, retrieval and update of particular facts. However, what constitutes the most efficient computer configuration of particular facts does not necessarily constitute something that corresponds to the configuration of the real world.

Information Engineering uses entity-relationship modelling as part of the methodology for relational database design. This is a tool for the analyst. It is not something that laymen can easily understand. Indeed the those people that are good at entity relationship modelling are usually those that are familiar with the operation of a relational database. Unlike conceptual models, and to a lesser extent data flow diagrams, are not intended to be part of the client interface.

Data flow diagrams represent a stylized view of the physical world and predicate logic could be said to represent valid arguments. It is difficult to see what entity-relationship models represent because relations, like propositions, do not exist in the real world. Relational database designers, however, speak of them as if they do.

The construction of normalized relational database does not require entity-relationship modelling it can instead be constructed from an empirical model such as that represented in figure 29. The advantages of using logico-linguistic modelling techniques are threefold: the models can be understood by the clients and therefore they stand a better chance of understanding the computer system; the models provide a more powerful and more standard logic upon which to base the design; the model has an explicit real world connection.

7.5.2 Terminology

The theory behind normalized relational database design is expressed in set theory (Codd, 1970). Set theory could be loosely described as the equivalent of predicate logic and even cautious authors note "strong affinities to logic" (Haack, 1978, p. 6) However, converting a model in predicate logic into third normal form involves a number of difficulties. The first of these concerns the vocabulary that database designers use to describe their work. Database Terminology (let us call it "DBT") borrows words from logic but gives them completely different meanings. This can be seen in the following table:

Logic	Database Terminology (DBT)
Instantiated object	Value

Table of relations	Relation
Predicate	Attribute, data item, domain, field.
Object	Entity, tuple
Relation	Set of tuples
Domain	No term
Truth value	No term

The difference in terminology might seem trivial but DBT is so bizarre that it is suspicious. It is to be remembered that the terminology of logic tries to remain as close as possible to the terms used in English grammar and to ordinary English usage. DBT appears to be trying distance itself from logic and ordinary usage. Whatever the reasons for the emergence of DBT its effect is to provide a smoke screen that obscures some dubious logical maneuvers.

7.5.3 Predicates into objects

It is interesting to note that three or more placed predicates are quite common in relational database design but comparatively rare in ordinary logic. Relational database designers like to turn predicates into objects. For example, if we wanted to say that x is a proposal that is made *quickly* by Ministry Official z, then in logic we would just define a new predicate:

Tz,x: x is a proposal that is made *quickly* by Ministry Official z

But a database designer would tend to define a new object:

Rz,x,y: x is a proposal that is made by Ministry Official z at speed y

The reason they take this counter intuitive approach is technical. Computers can process an inquiry more quickly if there are a small number of relationship tables containing a large number of objects than if there are a large number of relationship tables containing a small number of objects.

The maneuver of turning predicates into objects would be quite innocuous if it was undertaken as a deliberate bodge. It would be satisfactory if one took a model like figure 29 converted it into 3rd normal form, undertook operations on the database and then converted the results back into the form of the original model. But this does not appear to be what happens. Relational databases are built out of entity relationship models. These models which for database designers have acquire the status of representations of the world have the peculiar relational logic built into them.

7.5.4 Ontology of the object

Ockham's razor is a principle of ontological economy. It is stated as "Entities are not to be multiplied beyond necessity". Relational database design clearly employs an opposite principle.

Quine's theory of predicate logic states that objects in correctly formed quantified statements carry an ontological commitment (see Haack). That is, they refer to past, present or future, physical objects or events. As objects in database design are created as and when the designers choose, relational database design is incompatible with these ideas. Quine's theory is contentious but it is a major contender to explain truth and meaning.

Relational database design is, therefore, far from neutral with regard to ontology, the theory of meaning and the theory of truth. This is disturbing because writers on the subject of relational databases seen quite unaware of this fact. The problem comes home to roost when we consider time. Normally temporality will be expressed as a relation between events. Expressing temporal relations in the predicate calculus is not a serious problem. However, for the relational database designer time becomes an object that stands in some sort of relation to an event. Ideally a relational database requires an infinite number of time objects ready to stand in a relation to any new event. In practice only those time objects relevant to event objects are stored in the system. However, database designers have a problem deciding which and how many time objects are relevant to an event. This is one of the reasons that relational databases need to go through the troublesome process of being validated; a process that would be unnecessary if they were based on normal predicate logic.

7.5.5 Deriving a relational database

There are three ways our model can be used in the design of a relational database. The first is to take a model such as figure 29 and express it in a form of the predicate calculus that uses a minimum of relations, this will not be considered. The second is to use a conventional predicate formulation, the one given in section 7.4 above, and obtain a database that is not in third normal form. The third method is to take the database design obtained using the second method and convert it into third normal form.

If we which to make a relational database from figure 29, we need take only those bubbles on the outside of the figure. That is 16, 17, 15, 12, 13, 7 and 18. The internal bubbles (14, 10 and 11) are all logically equivalent to one or more of the outside bubbles but the outside bubbles taken together contain more predicates. Any instantiation in the set of predicates in the outside predicates will enable us to deduce any instantiation in the set of internal predicates. The internal predicates are, as far as an information processing system is concerned, redundant. In database jargon this means that attributes that are functionally dependent on other attributes have been removed.

It is a good idea to give each instantiation of an object variable a unique description in the relational database. This can be achieved by giving every instantiation a unique number. An artificial predicate is, therefore, created for every object variable. As there are only two objects in our model, i.e. x and z, only two artificial predicates need to be created. These are "Proposal number" and "Official number". These unique predicates (called key attributes in DBT) can then be substituted for all the ordinary predicates. A subject - predicate structure is called a "relation" in DBT even if the subject - predicate structure is only one-placed. This is in stark contrast to logic where the term "relation" is only used for a two or more placed predicate.

Thus when the substitution of key attributes is made for ordinary predicates we will find that a one-placed predicate has one key attribute, a two-placed predicate has two key attributes, and an n-placed predicate will have n key attributes.

With an asterisk to denote the key attributes figure 21 can be laid out as follows:

PROPOSAL

* Proposal number

  proposal name

OFFICIAL

* Official number

  Official name

PROPOSAL REGISTERED

* Proposal number

  Proposal name

OFFICIAL THINKS ETHICAL

* Official number

* Proposal number

OFFICIAL THINKS PERSONAL GAIN

* Official number

* Proposal number

OFFICIAL IN POSITION TO MAKE PROPOSAL

* Official number

* Proposal number

SANCTIONED BY MINISTER

* Proposal number

APPROVED BY CABINET

* Proposal number

PROPOSAL PASSED

* Proposal number

Here our objects have become the first two relationship tables and our predicates make up the rest. In this form the database is logically and ontological sound.

The move to third normal form is made by eliminating the relationship tables with only one attribute and making the attribute a truth function in a table that share the same primary key:

PROPOSAL

* Proposal number
  proposal name
  sanctioned by minister Yes/No
  approved by cabinet Yes/No
  proposal passed Yes/No
  proposal registered Yes/No

OFFICIAL

* Official number
  Official name

OFFICIAL THINKS ETHICAL

* Official number
* Proposal number

OFFICIAL THINKS PERSONAL GAIN

* Official number
* Proposal number

OFFICIAL IN POSITION TO MAKE PROPOSAL

* Official number
* Proposal number

This move is motivated by considerations of efficiency. Considering the ontological commitment of objects in predicate logic the result is quite bizarre because truth and falsity, represented by "Yes/No", have become objects.

7.5.6 Disadvantages of relational databases

The derivation of a relational database design from the empirical model, such as that expressed in figure 29, looses the modality and is, therefore, an extremely inelegant solution. Once modality is lost the problem of real world mapping once again begins to appear.

Relational database design is primarily a method of automating a bureaucracy. Designers generally use office forms as their main input for entity-relationship models. A paper based bureaucracy is itself an information system that may or may not map onto the real world. Like a paper based bureaucracy a relational database will only map onto the real world if it is part of a wider human activity system that can effect the mapping. Any falsified inductive hypotheses that are implicit in the relational database may be detected and removed during software maintenance. In this case software maintenance will be as vital a part of the information system as the software itself. Effective maintenance will involve tacit modal distinctions.

It is significant that Codd's seminal work (1970) begins "Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation)". On the surface this would seem very different from the emphasis on client led design that is in vogue today. However, few people would demand that clients and users must understand machine code. What is desirable is that the clients understand, and determine, the assumptions about the organizational context that systems analysts, database designers and programmers make in order to produce this code. The internal logic of the machine should reflect the wider logic of the human activity system.

In cases where, say the amount of data, makes a relational database the only practical solution, logico-linguistic modelling might not only provide a basis for design but also a blue print that could make explicit the modal distinctions needed in the maintenance of the software.

7.6 Another method for converting into Prolog

7.6.1 Problems with Prolog

Chapter 6 was concerned to quickly demonstrate the potential of a method. It was demonstrated how one model could be converted into Prolog. It also showed how the problem of converting biconditionals could be solved. However, the move from modal predicate logic to Prolog is not always as smooth or as simple as it appeared in that chapter. There are difficulties in this procedure that must now be considered.

Although we can derive a Prolog program from any formula in predicate logic, in many cases we cannot derive a Prolog program the is the equivalent of a formula in predicate logic. That is, if we take "PL" is stand for statements in predicate logic and "PO" to stand for statements that will run in Prolog, then PO -> PL will hold for all statements (every statement that can be expressed in Prolog can be expressed in predicate logic). However, PL -> PO will not hold for all statements (some statements that can be expressed predicate logic cannot be expressed in statements that will run in Prolog). Therefore, PO <-> PL cannot hold for every statement.

The "Dictionary" given in the appendix shows that many formulas in predicate logic can be expressed in alternative ways. These alternatives cannot be combined in a single program without the program entering into a vicious circularity. Given the same set of facts each alternative will be capable of queries that the other one cannot. Thus each alternative has only part of the power of the formula in predicate logic.

For example (Ax) (Rx & Tx & Bx) -> Px can be expressed in Prolog in such a way that it will maximize information about Px or in such a way that it will maximize information about Rx, Tx and Bx (see Appendix 11.3.2.4). This means that we must determine what we need information about and what information is going to be available before deciding which way to write the Prolog.

The remainder of this chapter describes another way of converting modal predicate logic into Prolog. In Chapter 6 the biconditional was handled by introducing "not" predicates and "not" objects. In this chapter the biconditionals will be converted into two sets of formulas each representing one half of the biconditional; these formulas are then converted into horn clauses with a positive antecedent, the Prolog rules follow from these.

The Dictionary shows two ways mutually exclusive, and none equivalent, Prolog expressions for most of the logical connectives used in logico-linguistic models. One logical connective is shown to have three exclusive Prolog expressions. The Dictionary makes no claim to completeness there may be many others. Thus there is no optimal way of expressing logico-linguistic models or even predicate logic in Prolog. The best way of writing the Prolog will depend on what questions it will need to answer. This will not be a problem in traditional applications such as deciding whether to approve or reject a request for a loan. Nor will it be a problem when the type of data available is always the same as in making deductions from sales figures.

The problem arises in the construction of a general model in which the questions that will need to be answered and the type data available are not known or will vary from time to time. Such a model can be constructed in using logico-linguistic models and expresses in modal predicate logic, but it cannot be fully expressed in Prolog.

7.6.2 Converting the logic into horn clauses

We will try to avoid the problems referred to above by writing two Prolog programs. One will contain a direction of implication going one way the other will be the opposite.

The formulation of the model in modal predicate logic was given in section 7.4. We have four main universals to contend with:

L (Ax) (Az) (Kx & Hx & Ix) <-> (Rz,x & Kx & Ix)

L (Ax) (Az) Rz,x <-> (Oz,x & Nz,x)

L (Ax) (Kx & Ix) <-> (Lx & Mx)

M (Ax) (Az) Nz,x <-> (Pz,x v Qz,x)

Program 1

L (Ax) (Az) (Rz,x & Kx & Ix) -> (Kx & Hx & Ix)

L (Ax) (Az) (Oz,x & Nz,x) -> Rz,x

L (Ax) (Lx & Mx) -> (Kx & Ix)

M (Ax) (Az) (Pz,x v Qz,x) -> Nz,x

Program 2

L (Ax) (Az) (Kx & Hx & Ix) -> (Rz,x & Kx & Ix)

L (Ax) (Az) Rz,x -> (Oz,x & Nz,x)

L (Ax) (Kx & Ix) -> (Lx & Mx)

M (Ax) (Az) Nz,x -> (Pz,x v Qz,x)

We will concentrate on Program 1 for the moment. The first thing to do is to convert it into horn clauses. The result is:

L (Ax) (Az) (Rz,x & Kx & Ix) -> Kx

L (Ax) (Az) (Rz,x & Kx & Ix) -> Ix

(These two horn clauses are true but trivial and can be dropped from the program. As a conditional is only false when the antecedent is true and the consequent false, these clauses must always be true. The logical import of the biconditional from which they were derived is only felt in program 2)

L (Ax) (Az) (Rz,x & Kx & Ix) -> Hx

L (Ax) (Az) (Oz,x & Nz,x) -> Rz,x

L (Ax) (Lx & Mx) -> Kx

L (Ax) (Lx & Mx) -> Ix

M (Ax) (Az) (Pz,x v Qz,x) -> Nz,x

As the last formula is an inductive hypothesis we need to include rules that will falsify it given counter indicating particulars. These will taken from the definition L (Ax) (Az) Nz,x <-> Sz,x. Thus we can say that the hypothesis *M* (Ax) (Az) Nz,x <-> (Pz,x v Qz,x) will be false if Pz,x & Qz,x & Oz,x & -Sz,x.

7.6.3 Horn clauses into Prolog (Program 1)

The logical predicates and the predicates used in the program will be as follows:

Hx: proposal_for_repeal (symbol)

Ix: proposal_passed (symbol)

Kx: proposal_made_at_ministry (symbol)

Lx: proposal_sanctioned_by_minister (symbol)

Mx: proposal_approved_by_cabinet (symbol)

Rz,x: proposal_made_official (symbol,symbol)

Oz,x: official_in_position_to_make_proposal (symbol,symbol)

Nz,x: official_willing_to_make_proposal (symbol,symbol)

Pz,x: official_thinks_proposal_ethical (symbol,symbol)

Qz,x: official_thinks_proposal_personal_gain (symbol,symbol)

Sz,x: proposal_registered (symbol,symbol)

The rules for Program 1, will therefore be:

```
proposal_for_repeal (X) if
proposal_made_official (Z,X) and
proposal_made_at_ministry (X) and
proposal_passed (X).

proposal_made_official (Z,X) if
official_in_position_to_make_proposal (Z,X) and
official_willing_to_make_proposal (Z,X).

proposal_made_at_ministry (X) if
proposal_sanctioned_by_minister (X) and
proposal_approved_by_cabinet (X).

proposal_passed (X) if
```

```
proposal_sanctioned_by_minister (X) and
proposal_approved_by_cabinet (X).

official_willing_to_make_proposal (Z,X) if
official_thinks_proposal_ethical (Z,X) or
official_thinks_proposal_personal_gain (Z,X).
```

The falsification criterion can be formulated in two rules as follows:

```
incorrecthyp (official_willing_iff_ethical_or_personal) if
official_thinks_proposal_ethical (Z,X) and
official_thinks_proposal_personal_gain (Z,X) and
official_in_position_to_make_proposal (Z,X) and
not (proposal_registered (Z,X)).
```

We can make up the following instantiations:

```
proposal_for_repeal (prop1).
proposal_for_repeal (prop2).
proposal_passed (prop1).
proposal_made_at_ministry (prop1).
proposal_sanctioned_by_minister (prop1).
proposal_sanctioned_by_minister (prop3).
proposal_sanctioned_by_minister (prop6).
proposal_approved_by_cabinet(prop1).
proposal_approved_by_cabinet(prop6).
proposal_made_official (smith,prop1).
proposal_made_official (smith,prop3).
proposal_made_official (jones,prop6).
```

```
official_in_position_to_make_proposal (smith,prop1).
official_in_position_to_make_proposal (smith,prop3).
official_in_position_to_make_proposal (smith,prop4).
official_in_position_to_make_proposal (jones,prop4).
official_in_position_to_make_proposal (jones,prop6).
official_willing_to_make_proposal (smith,prop1).
official_thinks_proposal_ethical (smith,prop1).
official_thinks_proposal_ethical (smith,prop3).
official_thinks_proposal_personal_gain (jones,prop6).
```

Given these and a nominal instantiation for "incorrecthyp" (in this case "nothing") Prolog will answer the following queries:

```
goal: proposal_made_official (Z,X) and
proposal_sanctioned_by_minister (X)
```

Turbo Prolog answers:

```
Z=smith, X=prop1
Z=smith, X=prop3
Z=jones, X=prop6
```

Which means that the Minister has sanctioned prop1 and prop3 made by Smith and prop6 made by Jones. A more detailed query might be:

```
goal: proposal_made_official (Z,X) and
proposal_sanctioned_by_minister (X) and
```

not (proposal_passed (X))

Turbo Prolog answers:

Z=smith, X=prop3

Which means only prop3 made by Smith has been sanctioned by the Minister but not passed.

Given a much larger system these types of query could be very useful, but it would seem that a similar result could be obtained from using a relational database and a standard query language. More interesting is the ability of the system to show that some of the hypotheses used in its construction are false. Given:

goal: incorrecthyp (X)

Prolog responds:

X=nothing
X=official_willing_iff_ethical_or_personal

The "nothing" is just Prolog returning the nominal instantiation. the important thing is that Prolog has told us that our hypothesis that officials who think that a proposal is ethical or that it will result in personal gain are not always willing to make the proposal. Prolog knows this because there is a case, i.e. Jones and Prop4, where an official was in a position to make a proposal that he thought ethical and would result in personal gain, but which was not registered. This could not have happened if he was willing to make the proposal.

7.6.4 Horn clauses into Prolog (Program 2)

Program 2

L (Ax) (Az) (Kx & Hx & Ix) -> (Rz,x & Kx & Ix)

L (Ax) (Az) Rz,x -> (Oz,x & Nz,x)

L (Ax) (Kx & Ix) -> (Lx & Mx)

M (Ax) (Az) Nz,x -> (Pz,x v Qz,x)

Converted into non-trivial horn clauses this is as follows:

L (Ax) (Az) (Kx & Hx & Ix) -> Rz,x

L (Ax) (Az) Rz,x -> Oz,x

L (Ax) (Az) Rz,x -> Nz,x

L (Ax) (Kx & Ix) -> Lx

L (Ax) (Kx & Ix) -> Mx

M (Ax) (Az) (Nz,x & -(Pz,x)) -> Qz,x

M (Ax) (Az) (Nz,x & -(Qz,x)) -> Pz,x

In Prolog these are:

```
proposal_made_official (Z,X) if
proposal_made_at_ministry (X) and
proposal_for_repeal (X) and
```

```
proposal_passed (X).

official_in_position_to_make_proposal (Z,X) if
proposal_made_official (Z,X).

official_willing_to_make_proposal (Z,X) if
proposal_made_official (Z,X).

proposal_sanctioned_by_minister (X) if
proposal_made_at_ministry (X) and
proposal_passed (X).

proposal_approved_by_cabinet (X) if
proposal_made_at_ministry (X) and
proposal_passed (X).

official_thinks_proposal_personal_gain (Z,X) if
official_willing_to_make_proposal (Z,X) and
not_official_thinks_proposal_ethical (Z,X).

official_thinks_proposal_ethical (Z,X) if
official_willing_to_make_proposal (Z,X) and
not_official_thinks_proposal_personal_gain (Z,X)).
```

Prolog's inability to handle negation correctly has forced us to introduce two new negative predicates these need to be declared as predicates:

```
not_official_thinks_proposal_ethical (symbol,symbol).
```

```
not_official_thinks_proposal_personal_gain (symbol,symbol).
```

We can give some instantiations:

```
not_official_thinks_proposal_ethical (smith,prop6).
not_official_thinks_proposal_personal_gain (smith,prop5).
```

The falsification criterion can be formulated the opposite way from the way it was formulated in Program 1. That is we can say that the hypothesis M (Ax) (Az) ((Nz,x) <-> (Pz,x v Qz,x)) will be false if Sz,x & Oz,x & -(Pz,x) & -(Qz,X). Thus:

```
incorrecthyp (official_willing_iff_ethical_or_personal) if
proposal_registered (Z,X) and
official_in_position_to_make_proposal (Z,X) and
not(official_thinks_proposal_ethical (Z,X)) and
not (official_thinks_proposal_personal_gain (Z,X)).
```

Prolog will now answer queries:

```
Goal: official_thinks_proposal_ethical (Z,X).

Z=smith, X=prop1
Z=smith, X=prop3
Z=jones, X=prop4
Z=smith, X=prop5
```

The fact that Smith thinks proposal 5 is ethical can be deduced from the fact that he has made the proposal official but does not think it will result in personal gain.

As it stands there is nothing to stop self contradictory data being entered into this program. We can enter:

```
official_thinks_proposal_ethical (smith,prop7).
not_official_thinks_proposal_ethical (smith,prop7).
```

This could be safeguarded against by including a new predicate:

```
incorrectdata(proposal_ethical) if
official_thinks_proposal_ethical (Z,X) and
not_official_thinks_proposal_ethical (Z,X).
```

A clause to this effect would have to be added to each predicate beginning with "not". The "incorrectdata(X)" goal could be run as a regular check on data entry.

PART III

8 SCIENTIFIC APPLICATIONS

8.1 Systems of modal logic

The connection between modal logic and science has already been made in section 5.4. However, chapters 5 and 6 were mostly concerned with how a language can be mapped on to the real world in a scientific way. The present chapter will be concerned to show that logico-linguistic models can be used in the process of scientific investigation. First some points about modal logic must be reconsidered.

The formal aspects of modal logic are both difficult and complex. The notation for, and the interpretation of, the propositional and predicate calculi may vary but both form unitary systems that are generally and uniformly accepted. There are different ways of axiomatizing the predicate calculus but they really amount to the same thing. That is, in system A we will have axioms that are theorems in B and in B we will have axioms that are theorems in A. It doesn't matter much which system of axiomatization we choose because what can be proven in one system is the same as what can be proven in another. The case is quite different with modal logics. There are dozens of different systems, each has different axioms and what can be proven in one is often very different from what can be proven in another.

Before using modal logic it would seem to be a good idea to specify which system is being used. Unfortunately a huge amount of work would be necessary to decide which system would be the most appropriate to the form of knowledge representation needed in section 6.3.4. The deduction of just one intuitively obvious conclusion required a lot of work and that needed to be backed up by the difficult logic in Appendix 2.

There is a deeper problem here because there are different ideas about what modality means. Chellas gives a semantic definition of modality:

> ...a proposition is *necessary* if it holds at all possible worlds, *possible* if it holds at some. (Chellas, 1980)

This is his only explanation of it. In chapter one of this book truth conditions are explained semantically in terms of formulas prefixed by the semantic turnstile (which can be translated as "is a semantic consequence of") rather than the syntactic turnstile "¦-" (which can be translated as "is deducible from").

Hughes & Cresswell are not so committed to possible word semantics:

> Among true propositions we can distinguish between those that *happen* to be true and those which are *bound* to be true ... A proposition which is bound to be true we call a *necessarily true* proposition ... one that is bound to be false we call an *impossible* proposition ... and one that is neither necessary nor impossible we call a *contingent* proposition. (Hughes & Cresswell, 1972, p. 22)

Hughes & Cresswell define modal notions in terms of logical necessity and logical contingency. They use the syntactic turnstile in their introduction of modal system "T".

In chapters 4, 5 and 6 a case has been made to show that stipulative definitions and factual statements can be distinguished by modal operators. The validity of this depends on how modal operators are interpreted. The interpretation given by Chellas is too narrow the one given by Hughes and Cresswell less so.

There are problems at the level of the interpretation of the formulas. Chellas gives $L(A) \rightarrow M(A)$ as a valid formula in S5 (actually he gives it as an exercise for students to prove). Suppose we take A to be "All panthers are black" then following Chellas' interpretation $L(A) \rightarrow M(A)$ will mean:

"If if it is true that panthers are black at all possible worlds then it is true that panthers are black at some actual world."

This makes sense. But if "*L*" is taken to mean true by definition and "*M*" is taken to mean is true as a matter of fact then we have:

"If it is true that all panthers are black by definition then it is true that all panthers are black as a matter of fact"

This creates epistemological difficulties. As was pointed out in section 4.5.1 if a statement is true as a matter of definition it cannot be established empirically - it cannot be established as a matter of fact. A similar problem occur in the modal system "T" where Hughes & Cresswell give the following rule:

$p \to M\, p$

Which would be interpreted as "if p is true then p is true as a matter of fact".

However, there are also a host of epistemological problems with possible world semantics. How can we establish that panthers are black in all possible worlds? We cannot, as a matter of fact, visit all possible worlds and it is doubtful if it is even logically possible to visit all possible worlds. Nor could we observe that panthers are black in a number of actual worlds and then formulate an induction to the effect that panthers are black in all possible worlds. An induction is always open to falsification by particular facts but any statement governed by the "*L*" modal operator is not.

There are more problems for possible world semantics when we come to consider extension and intension. The extension of "logically true" must be "at all possible worlds" but this cannot be an extensive definition because the members of the class of possible worlds has not been specified. Therefore, if "at all possible worlds" is to be defined at all it must be given an intensive definition. That is, if we are going to say that X is true in all possible worlds, then, because we cannot specify all the worlds in which it is true, we must have a criterion for judging that it is true in all possible worlds; and surely that criterion will be what "at all possible worlds" means. But possible worlds semantics does not provide such a criterion.

Although a formalization of the method advocated in this thesis could be undertaken in one of the existing modal systems, it is doubtful whether this would be worthwhile. It might be better to develop a new system of modal logic that includes a new interpretation of the modal operators. This in turn might require a new theory of truth.

Such a theory of truth could be constructed out of elements of Wittgenstein's rule based language games and Tarski's distinction between object language and meta-language. Stipulative definitions could be taken as meta-linguistic statements about an object language that refers to the real world. Statements in the object language would be true or false in so far as they correspond to the real world. They could would conform to the correspondence theory of truth and be denoted by the "*M*" modal operator. Statements in the meta-language would not refer to the object language but would be statements of the rules of the object language. They would conform to a coherence theory of truth and be denoted by the "*L*" modal operator.

This line of thought will be taken a little further in chapter 10. The use of SSM in information system design has hitherto lacked any formal basis. Other systems of information system design do not make modal distinctions. While the case for modal logics is a good one a single system has yet to be universally accepted and this indicates that existing systems are contentious. Possible world semantics is one attempt at basing modal logic in a theory of truth but this is also contentious.

Enough attention has been paid to formalism and highly abstract theory. In the next section an informal system of modal rules will be put forward and worked through as a practical tool in the design of a scientific knowledge based system. It might be possible to formalize this system but the work required is far beyond the scope of the thesis.

8.2 Informal rules for a modal system

The operator "*L*" will be used to denote a stipulative definition or a statement that is deducible from one or more stipulative definitions. The operator "*M*" will be used to denote a statement that is not a stipulative definition nor deducible from one or more stipulative definitions. We will call these "contingent". The rules of formulation and production and the axioms will be the same as for the predicate calculus to these will be added the meta-rules from section 6.3.4 and the following additional rules:

A1 All particular statements are contingently true or contingently false.

A2 All statements that are necessarily true or necessarily false are universals.

A3 Any conclusion derived from a set of statements that are all necessarily true is necessarily true.

A4 Any conclusion that is derived from a set of premises that are not all necessarily true is contingently true.

A5 All axioms are necessarily true.

A6 If the disjunction of a necessary statement and a contingent statement is false then the contingent statement will be false.

A7 If the disjunction of a universal contingent statement and a particular contingent statement is false, for example -((Ax) Fx v (Ex) Gx), then the universal statement will be false.

The justification for these rules is as follows:

A1: Particular statement are prefixed by the existential quantifier i.e. (Ex). This indicates that at least one x exists. Existence is contingent and, therefore, all particulars are contingent.

A2: All statements are either particulars or universals, therefore, as all particulars are contingent, if any statement is necessarily true it must be a universal.

A3: In the absence of contingency, necessity remains by default. Thus necessary premises can only lead to a necessary conclusion.

A4: If a statement is contingent it cannot be true or false by logical necessity. Therefore, any statement derived in whole or in part from contingent statements cannot be true or false by logical necessity.

A5: Axioms are regarded as necessary in order to distinguish them from other premises which might be contingent.

A6: The truth of axioms and the theorems derived from them must be preserved at all cost. Contingent universals are hypotheses and can be shown to be false without falsifying the whole system.

A7: Contingent universals are hypotheses about what can be observed, contingent particulars can correspond to what is actually observed.

This system of logic will be called "NC". A system generated using these "NC" will be consistent. Which means that no thesis (axiom, premise or theorem) will be the negation of any other thesis. It is not claimed that "NC" is complete in the sense that every well formed formula can be derived as a thesis. In fact it is doubtful that "NC" could be called an axiomatic system. It is more of an axiomatic shell in which premises can be added as required. It is not claimed that "NC" can deal with all the coherent statements of a natural language. Thus it is not intended to resolve many points of philosophical interest. It is only intended to deal with the statements produced in a logico-linguistic conceptual model.

8.3 A Scientific example

8.3.1 Modelling an inductive hypothesis

A scientific relationship between definitions and inductions can be clarified by means of an example, let us take *g* from figure 22. To build the figure the first requirement is to get the stake-holders to establish a defining criterion for "a patient has measles". In this example "*n* virus alpha in patient blood stream" and "*k* Alpha antibodies are not in patient blood stream" are taken as a necessary conditions of a patient having measles. The conjunction of *n* and *k* are taken as an N&S condition of a patient having measles. As this is a defining criterion it is flagged with "*L*" in the figure.

Here *n* and *k* are clearly intensive defining criteria for *g*. The extension for e will be all the members of the class of patients who have measles. In Table 1 individual members of this class are listed: Adam, Betty, Colin, Dianna etc.

Formally we can represent these defining criteria as

Domain: Hospital patients

Gx: x has measles
Nx: x has virus alpha in the blood stream
Kx: x has no alpha antibodies in the blood stream

Axiom (1) L (Ax) Gx <-> (Nx & Kx)

In other words we shall take it that any hospital patient that has measles has virus alpha and no alpha antibodies in their blood stream. A number of theorems follow from this:

Theorem (2) L (Ax) Gx -> (Nx & Kx)

from (1) by material equivalence, simplification and A3 or meta-rule one.

Theorem (3) L (Ax) Gx -> Nx

From (2) by material implication, distribution, simplification, material implication and A3 or meta-rule one.

Theorem (4) L (Ax) Gx -> Kx

From (2) by material implication, distribution, simplification, material implication and A3 or meta-rule one.

Theorem (5) L (Ax) (Nx & Kx) -> Gx

from (1) by material equivalence, simplification and A3.

Given this we can begin to formulate inductive hypotheses. The inductive hypothesis shown in figure 22 is that every patient that has measles has a runny nose, a high temperature, inflamed eyes and a skin rash, and that every patient that has a runny nose, a high temperature, inflamed eyes and a skin rash has measles. The formal expression is:

Fx: x has a runny nose

Hx: x has a high temperature

Ix: x has inflamed eyes

Jx: x has a skin rash

Prem (6) *M* (Ax) Gx -> (Fx & Hx & Ix & Jx)

Prem (7) *M* (Ax) (Fx & Hx & Ix & Jx) -> Gx

Theorem (8) *M* (Ax) Gx <-> (Fx & Hx & Ix & Jx)
from (6) and (7) by Material Equivalence and A4.

The biconditional, which is indicated by the symbol "<->", is sometimes known as identity. What (8) says in effect is that having measles is the same thing as having a runny nose, a high temperature, inflamed eyes and a skin rash. However, (8) has an "*M*" modal operator which means that this identity is contingent not logical as it is in Axiom (1); it is not a definition and is claimed to be true only as a matter of fact.

Other inferences are:

Theorem (9) *M* (Ax) (Fx & Hx & Ix & Jx) -> Nx
from (3) and (7) by Hypothetical Syllogism and A4.

Theorem (10) *M* (Ax) (Fx & Hx & Ix & Jx) -> Kx

from (4) and (7) by Hypothetical Syllogism and A4.

8.3.2 Incomplete information

Figure 22 is a useful model because it enables us to make inferences on the basis of incomplete information. The presence of virus alpha is a necessary condition of measles but it is not a sufficient condition. This means that it is possible for a patient to have virus alpha present but never develop measles. Therefore, testing for virus alpha can not tell that a patient has or will get measles, but it can tell us that the patient has not got it.

The same is true of independently testing for alpha antibodies. The absence of alpha antibodies will not tell us that the patient has measles but the presence of alpha antibodies will tell us that the patient does not have it.

It is only when we have the results of both tests that we can say that the patient has measles. Thus, in Table 1 we can be sure that Adam has measles but we cannot be sure that Betty has.

With measles the runny nose, high temperature and inflamed eyes are the first symptoms followed a couple of days later by the rash. Now from Table 1 we know that Dianna has measles without the results of a blood test because she has all the symptoms. We know this because of Prem (3) above. With Harry we cannot tell if he has measles because he may develop a rash later. In spite of the fact that Colin has the same symptoms as Harry we know that he has not got measles because he does not have virus alpha; this follows from Axiom (1).

8.3.3 Causal Consequences

The model's ability to deal with incomplete information is of comparatively minor importance when compared to the fact that it shows how measles can be cured.

As the state of affairs that makes n true is a necessary condition of g it follows that if we can bring about a state of affairs that makes n false, then g will also become false. That is, if we can bring it about that there is no alpha virus in the patient's blood stream then we will bring about the patient's not having measles. However, it must be understood that this is more of linguistic move than a way to alleviate a real world malady. The absence of virus alpha just means that whatever is wrong with the patient we are not going to *call it* "measles".

The real world significance lies in the contingent identity between g (measles) and f & h & i & j (the conjunction of the four symptoms). If this identity holds then the absence of virus alpha will mean that the patient cannot have all four symptoms, *ergo* the elimination of the virus will eliminate at least one of the symptoms.

8.3.4 Falsification

We are entitled to say that the model is scientific in the Popperian sense (Popper, 1992) because it could be falsified by conceivable real world events. Suppose we find a patient, Isabel, who has the four symptoms, *f* & *h* & *i* & *j*, but does not have virus alpha in her blood stream. This can be expressed as:

Prem (11) *M* (Ex) Fx & Hx & Ix & Jx & -Nx

This is incompatible with Theorem (9), so by rule A7, Theorem (9) is false; but the fact that (9) is shown to be false allows us to assert its negation, this becomes theorem (14).

Prem (11) is also incompatible with either Prem (7) or Theorem (3) from which Theorem (9) is derived. So by rules A6 and A7, Prem (7) is false; and this means that its negation is true this becomes theorem (12).

Theorem (8), which implies Prem (7), must also be false; so its negation becomes theorem (13).

Although Theorem (10) has not been shown to be false it can no longer be derived from the axioms and premises and thus it no longer a theorem. With the false premises and invalid theorems removed the system is now as follows:

Axiom (1) *L* (Ax) Gx <-> (Nx & Kx)

Theorem (2) *L* (Ax) Gx -> (Nx & Kx)

Theorem (3) *L* (Ax) Gx -> Nx

Theorem (4) *L* (Ax) Gx -> Kx

Theorem (5) *L* (Ax) (Nx & Kx) -> Gx

Prem (6) *M* (Ax) Gx -> (Fx & Hx & Ix & Jx)

Prem (11) *M* (Ex) Fx & Hx & Ix & Jx & -Nx

Theorem (12) *M* -(Ax) (Fx & Hx & Ix & Jx) -> Gx

Theorem (13) *M* -(Ax) Gx <-> (Fx & Hx & Ix & Jx)

Theorem (14) *M* -(Ax) (Fx & Hx & Ix & Jx) -> Nx

Of the original system Axiom (1) and the Theorems derived from it remain unaffected as does Prem (6). This is still useful because it shows that measles is be a sufficient (but not a necessary) condition of the symptoms. This will prompt us to look for other conditions that are sufficient, see figure 23.

With the exception of Axiom (1) and its derived theorems the model would be further falsified by finding a patient, Johnny, who did have virus alpha, did not have alpha antibodies but did not have all of the symptoms, see Table 1. However, a reduced hypothesis that measles was a sufficient condition of a high temperature, inflamed eyes and a skin rash would still be tenable.

It is only if we found a person, Karen, that has virus alpha, no alpha antibodies and none of the symptoms that we can say that the real world mapping attempt for the conceptual model has been completely falsified.

This example could be written in Prolog but there is little point as it would not be substantially different from the previous example given in section 6.3.5.2.

8.4 The necessity of necessity

The question that now needs to be addressed is whether inductions of the type made in the previous section can be made without the definitions. The answer is that they cannot. The contention that a runny nose, a high temperature, inflamed eyes and a skin rash are an N&S

condition of measles is an inductive hypothesis. This means that it can be tested and the nature of a test demands that there is more than one logically possible result. With induction each new datum corresponds to a test. In the example the hypothesis failed the test because of the new data. But the new data could not have been included in the system without the definition. If the defining criteria in Table 1 were not present the hypothesis could not have been falsified. Instead the cases where the symptoms were not present would be judged to be cases where the patient did not have measles. An inductive hypothesis needs something to be tested against and this can only be provided by a definition.

An inductive hypothesis cannot be tested against another inductive hypothesis. If the "presence of virus alpha and absence of alpha antibodies" criterion of measles were an inductive hypothesis rather than a definition, then the new datum in figure three would not tell us which hypothesis was wrong. We would know that either the virus hypothesis or the symptoms hypothesis was wrong but we would not know which. Also no amount of new data would enable us to decide which.

The notion of measles could be abandoned and an induction could be made solely on the basis of the continuous concomitance of the particulars i.e. presence of virus alpha and absence of alpha antibodies always being accompanied by the symptoms. But this would just move the problem one level back. We would still need definitional criteria to determine that virus alpha was present and alpha antibodies absent.

However, it must be said that some inductions are possible without definitions for the simple reason that something kindred to induction is possible without language. Pattern recognition based on particulars is enough to justify certain expectations. A language is not required to observe that a runny nose, a high temperature and inflamed eyes is always followed by a skin rash. Even Pavlov's dogs were capable of induction in this sense. But inductions of this type would be limited to observation made by an individual subject. The understanding of reports of observations made by other subjects would require a definitional framework. Inductions at this level might have significance in connectionist work in artificial intelligence but they play little part in building a system based on existing human knowledge.

9 ADDITIONAL ARGUMENTS

9.1 Confusion arising from Hume and Quine

9.1.1 Philosophy's Holy Trinity

The kernel of modern Anglo-American analytical philosophy consists of the theory of meaning, the theory of knowledge (epistemology) and the philosophy of logic. Any area of philosophical inquiry will either be a sub-species of one of these, as the philosophy of science is a sub-species of epistemology, or will fall back on the three. These three areas of study are so closely interwoven that it is virtually impossible to gain an understand of one without an understanding of the other two. Any theory in one of these areas will inevitably imply certain theories in the other, it is, therefore, impossible to separate them completely.

One of the main arguments in this thesis is that SSM conceptual models are definitional. In chapter 3 the main weight of the argument drew upon the theory of meaning, chapter 4 drew mainly upon the philosophy of logic, the present section will largely be concerned with arguments drawn from the theory of knowledge. The main epistemological considerations have been left to this late section for two reasons: one has been ease of exposition; the other is that the epistemological arguments are, in this context, the least powerful. The arguments put forward in this section were formulated before the Wittgensteinian interpretation of

SSM was developed. They return to issues concerning causation and to a certain extent arguments that have been in previous sections are repeated. However, these arguments are set in a slightly different context and may add to the clarity of the thesis.

9.1.2 Information theory and SSM

Two different approaches to information system design can be distinguished on the basis of information theory. One approach takes information to be physical the other takes it to be logical. Information as physical is closely associated with the information theory of Shannon (1948, Shannon & Weaver, 1949) though the theory itself is just concerned with the amount of information and is not committed to any ontological position. Here information is taken to be a signal of some sort and the amount of information in any given signal can be measured by Shannon's well known mathematical formula. Sayre (1976) who uses Shannon's theory as the basis for his philosophy of mind says "... information content may be an attribute of possible as well as actual events". But here we can take him to mean "physically possible" not "logically possible" that is possibility with an "M" modal operator not possibility with an "L" modal operator.

Carnap and Bar-Hillel (Bar-Hillel, 1953, Carnap, 1950) took a fundamentally different view, a view in which information is defined in terms of propositions or statements. The amount of information associated with any given proposition is, according to them, a function of the number of propositions that imply it and the number of propositions that it implies. A proposition will have minimal information if it is implied by many other propositions and it, itself, implies few. A proposition will have maximum information if it is implied by few other propositions and it, itself, implies many propositions.

If information is physical an information system design can be based on physical objects and events. Structured methods take this approach. The raw material is documents, behavior, messages and anything else that is associated with information and can be known empirically. A manual systems flowchart represents these raw materials in a systematic way. A data flow diagram is normally abstracted from a manual systems flowchart in much the same way as the stylized map of the London Underground is abstracted from a scale map of the underground rail routes. Computer scientists are apt to call data flow diagrams "logical" but, as traditionally produced, they are no more logical than the Underground Map. Shannon's theory suggests empirical methods of information system design. As all information is physical it can be understood by observation.

We can follow Kant (1890) and distinguish between an analytic truth and a synthetic truth. An analytic truth is one that is true by virtue of words and logic alone (a cat is a cat, for example) while a synthetic truth is true for other reasons. Carnap & Bar-Hillel's theory suggests an analytic method of information system design. From the foregoing chapters it should be clear that SSM belongs to the analytic rather than the empirical school of information theory. (This distinction will be explained in greater detail in the latter parts of this chapter).

Figure 35 compares the analytical and empirical approaches to information systems. SSADM follows the right hand route and is empirical throughout. The method advocated in this thesis follows the left hand route and is analytic until the empirical models are constructed. Wilson's method is to move from the Conceptual Model to information categories, then to the Maltese Cross. The Maltese cross with its information processing procedures and inputs and output has the same sort of connectives as data flow diagrams. However, the Maltese Cross is not *derived* (I use the word strictly here) from the Conceptual Models but is the product of the Conceptual Model plus empirical knowledge that enters the method at the information category stage. The problem with Wilson is that in his description of his method he does not acknowledge that the line is crossed, nor does he explain how the empirical input is to be obtained.

9.1.3 Three philosophical problems

There are three potential problems with regarding SSM models as analytic. The first is connected with information theory. It is the contention that if anything is analytic then it is tautological and if it is tautological it must be uniformative. We can accept that anything that is analytic is tautological but not that anything that is tautological is uninformative. The contention that tautologies are uninformative can follow from the Shannon/Sayre account of information but not from the Carnap/Bar-Hillel account. Previous chapters have attempted to show how tautologies can be informative but there are many other examples that indicate that they are.

The second is the problem of induction. This derives from an interpretation of David Hume, which will be shown to be mistaken. The problem of induction states that there is no rational grounds for induction. This contention if true would not invalidate the analytic account of SSM models, it would, however, invalidate the inductive hypotheses that were used in earlier chapters to explain how SSM models can be informative about real world event.

The third is more complex. It is the contention that if a logical structure is to map onto causality then causal sequences must have the same property of necessity that the logical structure has. In other words there must be causal necessity. This is a potential problem for three reasons:

(i) Some people, notably Hume, have held that there is no such thing as causal necessity.
(ii) The modal logic, as expounded in previous chapters, states that causal relations are always contingent not necessary.
(iii) Causal necessity would seem to imply determinism and the anti-reductionist stance of SSM is committed to indeterminism.

It will be argued that causal necessity and indeterminism are not incompatible. Modal logic can distinguish between logically necessary, logically contingent, causally necessary and causally contingent. Hume was wrong about causal necessity.

9.1.4 Conceptual models and tautologies

In a narrow sense it is correct to say that Conceptual Models are tautologies. Just as it would be true to say that arithmetic is a tautology. But this does not mean to say that arithmetic does not add to knowledge or that arithmetic is of no use when dealing with the real world - the same is true of Conceptual Models.

The formula "2 + 3 = 5" could be said to be tautologous, but it can still be used to understand the real world. If we take a box containing two billiard balls and then we throw three billiard balls into it then (presuming that there are no holes in the box and that billiard balls cannot reproduce themselves etc.) by taking the balls that were already in the box as an instantiation of "2" and the balls that were thrown in as an instantiation of "3" and applying the formula "2 + 3 = 5" we can determine that there are five billiard balls in the box. This can be verified by counting them.

In the same way some Conceptual Models will map onto the real world. But there is a difference here because not all Conceptual Models will map on. This needs to be made clear. If a Conceptual Model is correctly constructed and there are particular instantiations of every element that requires an instantiation then the Conceptual Model will map on. However, many Conceptual Models may be constructed where the required instantiations cannot be found and these, of course, will not map on. An example would be a model of a system to feed unicorns with ambrosia. This might be a properly constructed model, the problem is that we cannot find instantiations of unicorns or ambrosia.

The example need not be fabulous. We might model a system to purchase a fried chicken franchise for less than a thousand pounds. The model might be perfectly well constructed but it might turn out that there are no franchises for sale at less than a thousand pounds.

The point here is that arithmetic is one type of formal system that happens to map onto the world. There are many similar formal systems that do not (see the MU-puzzle in Hofstadter (1980)). My point is that it is not Conceptual Models *per se* that are analogous to a formal system but that each Conceptual Model is analogous to a formal system. Thus each Conceptual model may be isomorphic or not, just as each formal system may be isomorphic or not. To put it another way: the fact that something is a Conceptual Model does not mean that it will apply to the world just as the fact that something is a formal system does not mean that it applies to the world. In this sense it is contingent that a formal systems or a Conceptual Model will apply to the world.

There is another problem with my analytic account of Conceptual Models and this problem is connected to the worry about tautologies. Given that a *completed* Conceptual Models is analytic, that is, given that the elements follow tautologically from one an other, we can still ask the question how do we get a Conceptual Model? Are the building blocks of the conceptual models (the conceptual model equivalents of the premises, axioms and definitions of a formal system) analytic or synthetic? To answer this question we need to consider some basic philosophical distinctions.

9.1.5 Hume's fork

Hume's fork is a popular name for his fundamental distinction between propositions that state the relations between ideas and those that state matters of fact. This notion has been expanded to cover three ways of describing propositions:

a priori	*a posteriori*
analytic	synthetic
necessary	contingent

The distinction between *a priori* and *a posteriori* is perhaps the most simple, it is concerned with how things are known. Propositions, arguments or ideas can be known to be true *a priori* if they can be known by the mind alone. To know that a proposition is true *a posteriori* requires reference to experience. Effectively the more modern term "empirical" can be substituted for *a posteriori*.

Kant (1890), who originally made the distinction between analytic and synthetic, said that a statement was analytic if the predicate was contained in the subject, otherwise it was a synthetic truth. According to Flew's dictionary (entry under "analytic") this was later revised to two accounts which are non-equivalent:

"*2* A statement is an analytic truth or falsehood if it can be proved or disproved from definitions by means of logical laws..." which Flew's dictionary attributes to Frege, the logical positivists and Wittgenstein.

"*3* A statement is an analytic truth if it is true in virtue of the meanings of the words it contains..."

All three accounts agree that a synthetic statement is any true statement that is not true analytically.

It is not difficult to see why *2* and *3* are non-equivalent. The contention in this thesis is that Conceptual Models consist of stipulative definitions and the logical relations between them. By *3* they are completely analytic because *3* can include definitions. By *2* only the logical relations are analytic so the stipulative definitions themselves must be synthetic.

Which brings us to the distinction between necessary and contingent. Here there is no need for a technical account as these are commonly used concepts. Indeed, to offer a definition of these concepts would beg a question that be considered later when it will argued against Hume that there can be two types of necessity, causal necessity as well as logical necessity.

The conservative view (which Flew's dictionary attributes to the Logical Positivists, see entry under "Hume's fork") is that these types of statement form groups of three such that if a statement is known to be true *a priori* it must be analytic and necessary, whereas if a statement is known to be true empirically it must be synthetic and contingent. If the truth of a statement can be determined analytically, it must be necessary and can only be known *a priori*. If a true statement is determined synthetically, it must be contingent an can only be known empirically. If a statement is necessarily true then it must be analytic and can only be determined *a priori*. If a statement is only contingently true then it must be synthetic and can only be known empirically.

Now, by *2* above Conceptual Models would be synthetic. This is because they contain definitions and definitions are not analytic by *2*. Because the definitions are synthetic the whole Conceptual Model must be synthetic. In this respect the property of "being synthetic" functions like contingency in that it is transitive. To make this clear remember that in a deductive inference the conclusion will be contingent if one or more of the premises are contingent.

Also by *2* we can stick to the conservative view and say that Conceptual Models are contingent, but we can not stick to the conservative view and say that Conceptual Models must be empirical. People can think up conceptual models by themselves. And this is what happens, the facilitator builds a number of models, these are then put before the stakeholders to initiate a debate about what type of model is appropriate. To *know that* the stakeholders have agreed to a model might require empirical evidence but this is the same thing as saying that the agreement is *based* on empirical evidence. So, by *2* Conceptual Models are synthetic, contingent but can be built on the basis of *a priori* knowledge.

The conservative account can be maintained, however, if we decide that stipulative definitions should *not* be regarded as statements or propositions (until now I have used the terms propositions and statements interchangeably). This might seem like a good idea at first glance but there are problems with it. If stipulative definitions are not propositions then they cannot be true or false (propositions are truth bearers). And this means that they cannot function in a logical calculus. Take our favorite syllogism:

All men are mortal
Socrates is a man
Therefore:
Socrates is mortal

If mortal has been stipulated as a defining property of men then our first premise here could not be true (nor false). In which case, even if it was factually true that Socrates is a man, we would not be able to deduce the truth of the conclusion. What is just as bad is that if we are able to deduce the conclusion then we would have to be able to determine that the first premise was not a stipulative definition. This runs counter to common sense which suggests that we can make the above inference without considering whether the first premise is a definition or not.

Therefore, it is best to regard stipulative definitions as being true. To a certain extent this is arbitrary. But then a stipulative definition, like a name, is arbitrary.

If then we take it that stipulative definitions can have truth values and we accept *2* as a definition of "analytic" we must deviate from the conservative account. Deviating from the conservative account is by no means new. Kant held that synthetic *a priori* judgement are possible. More recently Kripke (1971) has argued that logically necessary truths can be known empirically.

The conservative position can, just, be maintained if we take *3* as the definition of analytic. Taking Conceptual Models as analytic we can agree with the conservative position that they can be built on the basis of *a priori* knowledge and that they are logically necessary. In fact knowledge hardly enters into it; if we take knowledge as being true, justified belief, then we could say that a stipulation justifies itself and that it is true (logically true) by virtue of its being formulated. This does, however, involve a rather unusual account of truth, this will be described in the final chapter.

9.1.6 The problem of induction

In order to answer the objection that logical relations cannot map onto causal relations, it will be necessary to show that there are two types of necessity i.e. logical necessity and causal necessity.

Hume held that there was no such thing as causal necessity: that experience neither reveals nor produces any necessity in the objects; that is, does not provide materials for any rational inference from cause to effect (or vice versa) in a new instance (Mackie 1974 p. 10). This view is connected with the notorious problem of induction. However, whether the argument that there is no such thing as causal necessity is a premise or a corollary of the problem of induction is more difficult to determine.

Before we can justify mapping logico-linguistic conceptual models onto real world causal sequences we must establish that causal necessity is possible. To be on the safe side let us assume that the problem of inductions gives some support to the idea that there is no causal necessity. Therefore, we must dispose of the problem of induction.

The problem of induction was put forward by Hume but it is not a simple matter to determine exactly what his argument was. Later writers emphasize different parts of what Hume had to say. It seems that two main arguments emerge. One is that induction is circular. The other is that induction is invalid.

In 1965 D. Stove produced an argument to the effect that not only did Hume fail to refute inductive probabilism he did not even consider it. Stove's paper is an historical piece but unlike most historical philosophical writings, which are of limited academic interest within the discipline, this paper has significant consequences for science. This is because, as Stove points out, later writers on the problem of induction refer back to Hume and assume that he had proven that there cannot be probable inductive arguments. It has been difficult to refute the problem of induction because it is difficult to find a coherent formulation of it and this is because a coherent formulation does not, and never did, exist.

Stove reconstructs, and expresses in modern terminology, Hume's writings on the subject:

> All *arguments from experience* presuppose that the future resembles the past (d). That the future resembles the past, however, is a *contingent* statement (e); consequently it is *not deducible from any premises (all of) which are necessarily true* (f), and if there

> are any arguments for this statement, they are *arguments from experience* (g). But - from (d) and (g) - any *argument from experience* for this statement would be circular (h). Hence - from (f) and (h) - no *predictive inductive inference* is one which reason engages us to make (i). (Stove, 1965)

While this might show that a deductive justification of inductive inferences is circular, it does not show that a probabilist justification of induction is circular. A probabilistic argument does not require the first premise (d).

To justify probabilistic inferences we do not need to accept that the future always resembles the past. All we need to assert in a probabilistic inference is that the future will probably resemble the past in some respects. If I say "it is probable that the sun will rise tomorrow" then nobody can accuse me of making a false statement if the sun does not rise tomorrow. The fact that the sun does not rise does *not* entail that it was *not* probable that it would rise. If I were assuming that the future resembles the past then I would say that the sun *will* rise tomorrow.

Mackie expresses essentially the same sentiment:

> Now Hume's premise that "reason" would have to rest on the principle of uniformity [the idea that the future resembles the past] holds only if it is assumed that

> reason's performances must all be deductively valid: if it were suggested that an observed constant conjunction of *A*s with *B*s *probabilifies* that this new *A* will be conjoined with a *B*, in terms of some logical or relational probability as proposed by Keynes and Carnap, that is, that some non-deductively-valid argument is none the less rational, that its premises really support though they do not entail its conclusion, then this possibility would not be excluded by Hume's argument, because such a probabilistic inference would not need to invoke the uniformity principle which produces the circularity that Hume has exposed. (Mackie, 1974, p. 15)

Some readers might think that there is a circularity in Mackie's argument because when he speaks of logical probability he seems to be smuggling in the concept of deductive validity by the back door. But this hinges on what you take Carnap's to have been doing in his "Logical Foundations of Probability" (Carnap, 1962). One might think that Carnap was trying to reduce all probabilistic statements to statements that can be established by deductive logic, just as Russell, in "Principia Mathematica" (Russell, 1910) attempted to reduce all arithmetical statement to statement that can be established by deductive logic. If this were the case then Mackie's argument would be circular and would not count against Hume's.

However, we need not interpret Carnap's work this way and we can be certain that Mackie did not (Mackie's works might not be the world best examples of lucid English prose, but he was a formidable logician and would not have been guilty of a simple circularity such as the one suggested above). We could take it that Carnap was trying to express probability in terms of standard logical constants and truth bearing variables rather than trying to effect a reduction of probability to the axioms of deductive logic. That is, the axioms of Carnap's system need not be taken as coextensive with the axioms of deductive logic.

This disposes of the circularity argument. The second argument is that induction is invalid. If "invalid" is taken to mean "not deductively valid" then this is certainly true. However, there does not seems to be any problem with invalidity in this sense unless it is assumed that the only reasonable inferences are valid inferences (deductive inferences). There is ample evidence that this is not so. Keynes took probability to be a primitive notion. In this case a reduction to another form of reasoning, deductive or otherwise, is unnecessary. Given this, we can effect a *reductio ad absurdum* against the deductive criticism of induction.

What we need to say is that some basic statement of probability theory (from Keynes, Carnap or somebody else) is self evident. Against this it can be argued that the claim of being self evident is not good enough and that proof is required, and further that the only form of proof available is deductive and a deductive proof cannot work without being circular.

Now let us challenge the deductivist by asking what justifies deduction. The first answer is that deduction is based on the law of contradiction, that $-(p \& -p)$, which is self evident. If we say that we do not agree that this is self evident, then the deductivist will claim that it is inconceivable that the law of contradiction is wrong.

Inconceivable can have two meanings. The first is logical, in this sense inconceivable means self contradictory. So, if the deductivist means inconceivable in this sense then what he is saying is that it is self contradictory to deny the law of contradiction. This can be expressed as $-(-(-(p \& -p)))$ which reduces to $-(p \& -p)$. In other words when you say "it is self contradictory to deny the law of contradiction" you are saying no more than what you are saying when you assert the law of contradiction. And this is not a good enough defense.

The second meaning of "inconceivable" is psychological. Here what we mean is that we cannot, as a matter of fact, imagine an example of something that violates the law of contradiction. No doubt this is true. Also we can find good evidence for the claim that nobody has ever been able to imagine something that violates the law of contradiction. But what evidence do we have that nobody will *ever* be able to imagine something that violates the law of contradiction. The deductivist cannot appeal to the logical sense of inconceivable because that says nothing. The deductivists only other option is to appeal to induction.

This works. As there as never, in the past, been a case where a person has been able to imagine something that violates the law of contradiction, there is good evidence that there will never be such a case. So, induction supports the law of contradiction which forms the basis of deduction (technically it might be better to say that induction supports the real world statement that corresponds to the law of contradiction, this is "an object cannot have a property and not have the same property simultaneously"). However, the deductivist cannot use this inductive argument because he claims that induction is unreasonable. The deductivist must, therefore, assert deductive methods without proof or evidence any yet simultaneously criticize induction for its lack of proof or evidence.

There are only two ways out of this dilemma. One is to accept both the law of contradiction and some basic statement of probability theory as being self evident. There other is to accept as self evident just the basic statement of probability; having accepted this the law of contradiction, and with the system of deduction, can be established inductively.

This leads to the odd state of affairs whereby the reductionist, who wants to establish his system with the minimum number of premises, must base his system on induction rather than deduction. Traditionally it has been the other way round. A plethora of philosophers from Decartes onwards have tried to base there systems purely on deduction. The fact that induction supports deduction but deduction does not support induction may provide a psychological explanation, but not a logical explanation, for them thinking that deduction was a stronger foundation.

The other important consequence is that the absence of a problem of induction firmly pulls the rug from under Popper's philosophy of science. And this in turn undermines the other new wave philosophers of science (such as Lakatos, Kuhn and Feyerabend) who were largely concerned with seeking a better solution than Popper's. This is not to say that everything in the new wave theories is wrong. The falsifiability criterion, used by Popper, can be justified (indeed it is one of the main principles used in this thesis), but it not justified because of a problem with induction.

This ties in with the discussion immediately above. The falsifiability criterion is the idea that any scientific hypothesis must be capable of being shown to be false. Of course, if a scientific hypothesis is correct it cannot be shown to be false. However, even if it is correct it must be logically possible (possible without self contradiction) for it to be shown to be false.

For example, if "pigs cannot fly" is a scientific hypothesis then there should be no self contradiction in saying "this is a pig and it can fly". So, given that there is no self contradiction here, it will always be possible (logically) to find a flying pig and show that the hypothesis if false. However, "pigs cannot fly" would be an analytic statement rather than a scientific statement if we included the inability to fly as a defining characteristic of pigs. In this case the statement "this is a pig and it can fly" would be self contradictory. Popper, rightly, claims that such things as psychoanalysis are pseudo-sciences because they are not scientific in this sense.

But there is a problem here. As logico-linguistic models are analytic, they are not falsifiable. So, how can they play a role that is not pseudo-scientific. The answer is that logico-linguistic models are analogous to formal system and like formal systems are not, in themselves, falsifiable or scientific. However, corresponding to them are statements to the effect "this formal system or this logico-linguistic model maps onto the real world". These statements are scientific and *are* falsifiable.

For, example, if there are instantiations for all the required elements in the gor tonking example of chapter 5, then the model will map onto the world. If we notice that the model has been continuously mapping onto the world for some time then we would be justified, by inductive reasoning, in stating that it will continue to map on (within a certain range of possibilities).

9.1.7 Hume and causal necessity

We now need to establish that causal necessity is possible. Hume was successful in demonstrating that causal connections cannot be logically necessary. This is easily demonstrated by the fact that even if it is true that *A* caused *B* we can still assert without contradiction that *A* did not cause *B*.

This is not Hume's only contribution to the theory of causation, he also demonstrated that the idea of *causal power* is false. In the eighteenth century it was widely believed that an efficient cause produced its effect by virtue of its power to do so (Taylor, 1963). Thus if one could observe this power in an object one would be given a warrant to make *a priori* inferences about what effects that object would bring about. The idea of causal power is apt to seem as strange to the twentieth century reader as the idea of a person's soul being located in his liver; but the reason it seems strange is because Hume disposed of the notion two hundred years ago.

Here, as with the problem of induction, Hume has often been credited with more than his due. Because he showed that causal necessity cannot be logical necessity and because there cannot be causal necessity arising from power, it is assumed that there cannot be any form of causal necessity. But as Mackie has pointed out Hume has *not* shown that there *cannot* be a necessity that distinguishes causal as apposed to non-causal sequences (Mackie, p. 12).

We can postulate a metaphysical necessity. We can say that the universe is such that events of type *A* always bring about events of type *B* and always will bring these about. We can say that blue litmus paper being immersed in acid necessitates its turning red. We can acquire evidence for these types of statement by means of inductive probability. The fact that there have been tens of thousands of cases where blue litmus paper immersed in acid turned red, and no cases where it did not, makes it highly probable that blue litmus paper being immersed in acid necessitates its turning red.

It should be noted here that using probability to justify causal necessity is quite different from a probabilistic theory of causation such as that developed by Suppes (1970). This is the idea that causal relations are essentially probabilistic. Thus, in saying that *A* causes *B* what we really mean is that *A* is very likely to cause *B*. Mackie shows that this idea is the result of a confusion:

> Saying that *A* is likely to cause *B* does not put ikelihood into the causing itself: it could mean ... that *A* is likely to necessitate *B*..." (Mackie, 1980, p. 50)

Modal logic is quite capable of making the distinctions that are required by the findings of this section. We need to make a distinction between logically necessary, logically contingent, causally necessary and causally contingent. Standard modal logics can handle this provided it is understood that what is causally necessary can be logically contingent. That is it may be causally necessary that the sun will rise tomorrow but we can deny that the sun will rise tomorrow without self contradiction.

9.1.8 Causal necessity and indeterminism

A fourth problem, that Hume was *not* responsible for, but which follows from the arguments immediately above, concerns determinism.

The foregoing has made a case for causal necessity although the arguments put forward in previous chapters could probably survive without this notion. As the anti-reductionist stance of SSM is contrary to determinism it is much less likely they these arguments could survive if determinism could be shown to be an outcome of their premises.

Therefore, causal necessity and indeterminism need to be reconciled. However, the apparent contradiction between causal necessity and indeterminism is only superficial. Indeterminism is merely the negation of determinism. Determinism can be stated as the idea that there are one set of necessary and sufficient condition for every event. A belief in causal necessity does not require us to believe that *every* event is paired with a certain individual set of necessary and sufficient conditions but only that *some* events are.

Mackie gives the example of an atom emitting random particles - the state of the atom is a necessary but insufficient condition of a particular particle being emitted at a particular time. Here the state of the atom cannot be said to necessitate the emission of the particle at a particular time.

But the existence of this sort of randomness does not entail that all events are random and indeterminate. We could take it that the randomness coming up from the quantum level averages out in most cases, therefore, we can have a set of necessary and sufficient conditions for most events. And we could also say that the randomness does not average out in all cases, an example would be the throw of dice where there are no conditions are sufficient to produce a given result such as two sixes.

9.1.9 Note on Quine

Kant and Kripke were cited above as philosophers who have argued that Hume's Fork can be bridged. Quine (1961) seminal paper "Two dogmas of empiricism" is more radical and argues that the analytic/synthetic distinction is spurious.

At the begining of his essay Quine shows that he understands the importance of definitions in the analytic/synthetic distinction. However, he recognizes only four types of definition. Lexicographical definition in which the lexicographer studies existing languages in order to find synonyms. Philosophical definition in which a recondite term is expressed by paraphrasing it into the terms of a more familiar vocabulary. Both of these are dependent upon a synonymy that existed prior to the exposition. A third type of definition is Carnap's *explication*. Here the definiens supplements the meaning of the definiendum. But Quine considers that this rests on *other* previously existing synonyms. The forth type is the introduction of novel notation for purposes of sheer abbreviation.

It is strange that nowhere in his essay does Quine mention *stipulative definition*. This is the type of definition that we have been mostly concerned with above. Stipulative definition does not require previously existing synonymy of any kind. To determine that *premeditated manslaughter* is *murder* a legislative body does not need to consult existing synonyms. It is hardly likely that Quine thought that a stipulative definition is a abbreviation. Consequences follow from stipulative definitions that do not follow from, say, making up an acronym. It is difficult to think of a case where someone was hanged because of an acronym.

Earlier in the essay Quine pointed out that meaning is not to be identified with naming. Is it possible that Quine would regard a case of stipulative definition as a case of naming?

Naming is the process of assigning a reference (extension, denotation, *bedeutung*) to a term, or assigning a term to a reference. Quine would seem to go along with this (1961, p. 21). A proper name has reference without having any sense (intention, connotation, *sinn*, and sometimes, confusingly, meaning). If we name the family cat "Tiddles" then nothing can be inferred from this except things to do with the name directly, such as "the family cat is called *Tiddles*" or "the name of the family cat begins with *T*".

We would actually want to say that naming is analytic. But in the case of proper names it is uninterestingly so, because no knowledge other than the name itself follows from naming. The case with general terms is more difficult because they tend to have sense *and* reference. The word "cat" has sense: from the fact that Tiddles is a cat certain things can be deduced such as the fact that Tiddles is a quadruped. This is part of the sense of the word "cat".

Now suppose I make a stipulative definition: "QChess is a game that is played by the same rules as chess except that the Queen can only move to a vacant square but can, in any move, move to any vacant square on the board". This clearly has sense because any chess player can readily begin to imagine what a game of QChess would be like. But QChess has no reference because a game of QChess has never been played.

To this it might be objected that QChess refers to a game that is in my head. And that what I have done is just to give a name to my idea of a funny game of chess.

There are all sorts of problems with this explanation, but let them pass. Whatever goes on when I made up QChess, be it a stipulative definition or a name of an idea, inferences can be made from it. We can infer, for example, that the Queen in a game of QChess can never take a piece. And this sort of inference can constitute knowledge. After all, the proposition *the Queen in a game of QChess can never take a* *piece* is a true, justified, belief. Further, we can say that this sort of knowledge is quite different from the type of knowledge that does *not* follow from stipulative definitions.

The finial objection that a disciple of Quine might want to make is that the *a priori*/empirical distinction covers the distinction that we want to make. But it doesn't because *a priori* knowledge is tied to a particular knowing subject. I say valid moves in chess are analytic because they follow from stipulative definitions. But I, personally, had to learn to make valid moves in chess by experience - by reading about chess and watching people play.

We need to distinguish between *things that people make up and what follows from them* and *things that are not made up and what follows from them*. This is essential to the analytic/synthetic distinction. Quine has not shown that it is invalid.

9.1.10 Truth and stipulative definition

One of the reasons Quine may have ignored stipulative definition is because traditionally it has been thought that they do not have a truth value. This is the view of Copi (1971) and Robinson (1954). That Copi gives little reason to justify this is understandable because he was writing a textbook which aimed at covering the whole range of traditional logic in a small space. Robinson, however, wrote an entire book on definition but his contention that stipulative definitions have no truth value is little more than a bland assertion. He states:

> A lexical definition is an assertion that certain people use a certain word in a certain way, and is therefore either true or false. A stipulative definition, however, is not an assertion at all. Therefore, since assertions are the only sentences that have truth value, it has no truth value. (Robinson, 1954, pp. 62, 63)

Robinson is not necessarily right here. A stipulative definition could be seem as containing an implicit assertion. For example, if in a piece of writing I stipulate that the word "cat" will refer to an animal with a tail, then I am, implicitly, making an assertion about how the word "cat" will be used in that piece of writing.

Robinson seems to miss the main problem about giving stipulative definitions truth values and that is that nothing implies their truth or falsity and nothing could count as evidence for their truth or falsity. However, in this respect they would not be unique. Self-referential statements such as "This sentence is in English" have the same status, they are implied by nothing but themselves and nothing but themselves counts as evidence for them.

Robinson's idea is that a stipulative definition "...is more like a *request* to the reader that he will understand the word in a certain way, or a *command*...". This is not good enough because there is never an inconsistency if a request is not taken up nor is there ever any inconsistency if a command is not obeyed, but there is an inconsistency if a stipulative definition is not adhered to. If I have stipulated that "cat" refers to an animal with a tail and then, in the same piece of writing, go on to say that Manx cats are an interesting species, then, clearly, I will have been inconsistent.

Robinson recognizes this and says "That stipulative definitions lack truth value does not prevent words stipulatively defined from being used to make true or false statements. If you stipulate that "nacks" is to mean roses and "braze" is to mean smell sweet, it is false that nacks never braze."

What needs to be asked here is how can we possibly say that it is false that nacks never braze unless we assign truth to the two stipulative definitions. We can work this one out.

Let F be the predicate "is a rose"

Let G be the predicate "is a nack"

Let H be the predicate "it smells sweet"

Let I be the predicate "it braze"

Now to say that it is false that roses never smell sweet is to assert:

-(Ax) -(Fx & Hx)

or the equivalent:

1 (Ex) Fx & Hx

Now given:

2 (Ax) Fx <-> Gx

3 (Ax) Hx <-> Ix

we can derive:

4 (Ex) Gx & Ix

but we can only do this if we are given *2* and *3*.

To put it in English, the falsity of "nacks never braze" can *only* be derived from the fact that some roses smell sweet if the statements "roses are identical with nacks" and "braze is identical with a sweet smell" are *true*. But Robinson has said that stipulative definitions have no truth value, therefore on his account, these identity statements cannot be true.

This brings out the dilemma with stipulative definition. While stipulative definitions do not follow from the truth or falsity of any other statements, we *would* like to say that the truth and falsity of other statements do follow from stipulative definitions. To get a clear grasp of what is going on here let us consider a set of stipulative definitions.

---- 0 ----

THE RULES OF QCHESS

Rule 1: The Queen in a game of QChess can only move to an unoccupied square.

Rule 2: The Queen can move to any unoccupied square.

Rule 3: Except for moves made under rules 1 and 2 normal chess rules apply

---- 0 ----

From the above we can make the following inference:

Inference 1: The Queen can never take a piece.

This inference follows from rule 1 and rule 3. However, if stipulative definitions are neither true nor false then we cannot determine the truth value of Inference 1 on the basis of the rules. We cannot say that Inference 1 is true or false. The most we can say that is true would be "Inference 1 follows from rules 1 and 3".

However, suppose we said:

Inference 1A: The Queen can never take a piece in QChess.

This is more likely to gain immediate acceptance than Inference 1 because there is the worry with Inference 1 that it could be taken to refer to normal chess. This brings out the point that stipulative definitions are only true in a Universe of Discourse. To put it simply Inference 1 is only true of QChess.

Stipulative definitions tend to work badly where the Universe of Discourse is not specified or understood. Also it is appropriate to note here that in logic the Universe of Discourse is identified with the domain of quantification. We can see how this ties together with Robinson's stipulation that roses are identical with nacks. This is true in the Universe of discourse that comprises Robinson's book. The formula (Ax) Fx <-> Gx could be translated as "for every x that is in Robinson's book Fx <-> Gx".

Previous chapters give reason for believing that stipulative definitions or the equivalent are a necessary part of any information system. From which it would seem to follow that every information system has a limited Universe of

Discourse. This leads to the bizarre idea that every predicative function in an information system is true only of objects in that information system. But this is not as far removed from reality as it might seem at first sight. Bureaucracies have a tendency to define their own entities and become largely self verifying. This was well illustrated in Franz Kafka's *The Castle* (1930). This cannot be avoided by the simple device of using lexical definitions instead of stipulative ones because natural languages themselves are based on stipulative definitions or similar ("naming" for example).

But to return to the main point. Although bringing in the notion of a Universe of Discourse helps, it does not entirely solve the problem of assigning truth values to stipulative definitions. All that we have established so far is that Inference 1A will be true if it is true that "Rule 1 and rule 3 are rules of QChess", however, this is a statement that can be true or false. What leads us to believe that Rule 1 and rule 3 are rules of QChess is the fact that they are written down on the bit of paper above.

What this amounts to is that the promulgation of a stipulative definition entails that the definition is true in a Universe of discourse. We could, therefore, argue that stipulative definitions are not just logically similar to self-referential statements, but that they are in fact one species of self-referential statement. A more plausible idea is that they are Tarski style meta-linguistic statements.

9.1.11 Note on fuzzy logic

Returning to the mistake made by Suppes (in section 9.2.5), we can note here that the same confusion arises in fuzzy logic or fuzzy sets:

> In Boolean algebra, 1 represents truth and 0 falsity. So it is in fuzzy logic; but in addition, all the fractions between zero and one are employed to indicate partial truth. Thus
>
> p(tall(X)) = 0.75
>
> states that the proposition that "X is tall" is in some sense three quarters true. It is, by the same token, one quarter false." (Forsyth, 1984)

Here there is no need whatsoever to deviate from the traditional two valued logic and generally accepted English usage. There is no need to introduce the counter intuitive notion of "three quarters true". We can just as easily say that it is three quarters probable that it is true.

But not only does fuzzy logic fail to improve on our existing concepts it is also likely to be dangerous as it stands to confuse various types of probability.

If by "X is tall" we mean that X is taller than other people then we can find ourselves in a situation where three quarters of the people are shorter than X and one quarter of the people are taller than X. Given this, if we take any person at random there will be a three quarters chance that X will be taller than they are. This is an absolute truth.

We can know this absolute fact by conducting a survey of the entire population, but in most cases this will not be possible. Therefore, we will have to make a statistical sampling of the population, but a statistical sampling does not allow us to say anything with certainty. On the basis of a 5% sample of the population we might be able to say that it is 90% probable that there is a 75% chance of any given member of the population being shorter than X. There are two different types of probability here and both are likely to change independently of each other. Firstly, we might conduct another survey of the population which will give us more evidence. But secondly, the population might change over time, with better nutrition more people would become taller than X.

No doubt fuzzy logicians will find a way of separating these two types of probability, but there is no point because traditional logic already separates them. Also we are likely to find that in order to make this separation fuzzy logician will have to smuggle two valued logic in through the back door.

9.2 Other types of knowledge representation

9.2.1 A plethora of methods

A fairly comprehensive account of the various methods of knowledge representation can be found in Ringland & Duce (1988). This includes accounts of dozens of different methods. Almost all of these have some features in common with logico-linguistic modelling and logico-linguistic modelling has some features which differ from them all. A full comparison of all of them in terms of commonality and divergence would take up far too much space and would in the case of most of the other methods be a tedious and largely pointless exercise.

Instead a brief comparison will be made with Sowa's conceptual graphs and with Minsky's frames. Both of these are interesting and both have some interesting similarities to logico-linguistic models. With both a full exploration of the points of divergence would very rapidly take us into discussions of philosophical issues that are beyond the scope of this thesis.

9.2.2 Sowa's conceptual graphs

Sowa's conceptual graphs have a bubble diagram style, are concerned with concepts and can be expressed in the predicate calculus; they are, therefore, superficially very similar to Logico-linguistic models. Like many other semantic net style graphs the logical and epistemological status of Sowa's graphs is not perfectly clear. He recently described them as "a graphic system of logic ... equivalent to predicate logic" (Sowa, 1992), yet earlier he described them as "a method of representing mental models" (Sowa, p. 4).

As a graphic system of logic Sowa's graphs differ from logico-linguistic models firstly, in that they do not include modality which is one of the principle features of logico-linguistic models. The second difference is that Sowa's graphs contain the plethora of detail needed to capture the vagaries of English syntax. For example, the verb "to run" is represented by ten bubbles and ten arrows in a conceptual graph (Nogier & Zock, 1992). Such detail is not essential for the construction of a knowledge based system nor is it practical as a stake-holder driven modelling device.

It is clear that conceptual graphs are a tool for an analyst intending to represent discourse in a natural language. Logico-linguistic models, by contrast, are not intended to represent a natural language but are intended to be an artificial language. Logico-linguistic models are not, therefore, dependent on lexicographical science nor are they prone to the paradoxes of self reference which are a feature of natural languages. It is pertinent to point out here that it may, as both Frege and Tarski believed, be impossible to formulate a theory of truth for natural languages (Grayling, 1990, p. 248).

If Sowa's conceptual graphs are intended to be mental models then it seems they are very different from Logico-linguistic models. Logico-linguistic models are based on the Wittgensteinian theory of language which contends that mental models, if they are anything other publicly observable neurological states or elements of a public language, simply do not exist. Sowa's discussion of "percepts" (1984, p. 24) sounds very similar the sense datum theories that were discredited by Wittgenstein's private language argument. Logico-linguistic models are not intended to be representations of mental models nor are they representations of anything, they are just records of an agreement to uses words in a certain way.

It seems that the theory of meaning which forms the basis of Sowa's graphs is fundamentally different from the one that is assumed in here. This could explain the fact that Sowa's graphs lack the modal operators that form vital components of empirical models described in previous chapters.

9.2.3 Frames

Structured object representation in terms of schemata and frames exhibits some similarities to logico-linguistic modelling. The "objects" have many similarities to the object variables of the predicate calculus. The theory behind frames is based on some weighty philosophical arguments. Non-monotonic logic is used with frames.

Frames originate with Minsky's (1975) attempt to represent common-sense thought. In this paper he encountered the problem of natural kinds. This problem goes, briefly, as follows: elephants have four legs, but an elephant can loose a leg without ceasing to be an elephant; also it would seem that an elephant could loose his other legs and his trunk and ears etc. and still be an elephant; so it would seem that anything we might use to define or describe an elephant might be lost without the animal ceasing to be an elephant; so, what is an elephant? Minsky's answer was default reasoning which is, briefly, the idea that an elephant has four legs, a trunk, ears etc. by default but a particular elephant may lack one or more of these attributes. This type of default reasoning was later formalized by McDermot & Doyle (1980) in a system of non-monotonic logic.

It could be argued that natural kinds is not a real problem and that it is a futile search of an absolute meaning for terms such as "elephant". Given Wittgenstein's language game theory we might argue that there is no absolute meaning. To follow through this line of thought would be a major digression into philosophy. In practice natural kinds does not seem to be a problem in SSM conceptual modelling. The problems that occur when trying to define "elephant" in an absolute sense do not seem to occur when defining "working elephant" in the context of a human activity system.

McDermot's non-monotonic logic is, therefore, designed to do a different job from the Popperian falsification for real world mapping task that is required of the non-monotonic logic used in this thesis.

9.2.4 Modal logic in knowledge elicitation

Modal logic is a powerful tool that has many uses. It is liable to turn up in a wide variety of artificial intelligence applications. It could easily be added on to one of the many schemes of knowledge representation. However, it does not appear to have been used, explicitly, in the process of knowledge elicitation. The most important distinguishing factor of knowledge representation models advocated in this thesis is the fact that they are an outcome of the explicit use of modal logic in the knowledge elicitation process.

PART IV

10 CONCLUSIONS AND RESEARCH DIRECTIONS

10.1 Results

The results of the research that is described in this thesis are many and varied. It has shown how SSM models can be expressed in formal logic and has provided an explanation of the meaning these models in terms of one of the twentieth century's most important theories in the philosophy of language. This in itself is no mean achievement. Philosophical writings on SSM have, with few exceptions, been confined to the theory of knowledge and SSM has lacked the logic and theory of meaning that is required for a comprehensive philosophical foundation. It has also introduced a degree of rigor, which has hitherto been rare, into the subject of SSM and philosophy. Hopefully it has supplied some of the scholarship that Checkland (1992) recently called for.

It has developed these ideas into a method for information system design. This method is supported by detailed argument that show why each stage in the method is necessary. It is a method based on extensive theory and can be contrasted with Wilson's method which is based on case examples. This does not detract from the value of Wilson's work, indeed, much of the work in the thesis was prompted by Wilson's work. The point is that the method and reasoning behind the method is quite different from Wilson. Both are of value in their own way.

The theoretical findings have opened up the possibility of a range of new application for SSM models. These have been detailed in previous chapters and include formulas for monitor and control, cause and effect analysis, a logical model of efficiency, logical mechanisms for real world mapping, knowledge elicitation, knowledge representation, knowledge based system design, scientific models, information requirements identification and relational database design.

In addition to the findings presented in this thesis, logico-linguistic model have prompted work by other authors. Merali (1993) has described how, firstly, logically enhanced models can be used to enrich the iterative debate and secondly, how the consensus that is reached at the end of the debate can be expressed in a logical model.

The finding of the thesis go beyond SSM and can have application independently of it. It has been shown that modal logic can be used in knowledge elicitation and how the results of this can be expressed in a non-monotonic form of Prolog. The thesis, therefore, makes a considerable contribution to cognitive science and artificial intelligence.

Finally it can be noted that the thesis gives something of a new perspective on the theory of truth, on the principles underlying modal logic and upon what would be require to formulate an inductive program. This ideas are detailed in section 10.3.

10.2 Action Research

Much of the writing on the philosophical aspects of SSM for information system design is of dubious value for two reason. Firstly, it is not grounded in the literature of an established discipline and is of questionable internal coherence. Secondly, because it is not clear how it could make a difference to the structure of an information system, it is not clear how it could be established that the use of these philosophical ideas has produced a system that is different from a system that would have been designed using ordinary methods.

This thesis is not open to these objections because the philosophical ideas work their way through to computer code. There can be no doubt that if the proposed methods were used to design an information system the resulting system would be significantly different from one designed by other methods.

The thesis contends that certain processes must be implicit in the creation of successful information systems. This argument is logical which means that, if the argument has been properly formulated, it cannot be denied without self contradiction. The thesis recommends that these implicit processes be made explicit in a certain sort of way and that contends that if this is done better sorts of information system can be created. This argument needs to be established empirically.

Empirical research will need to establish: a) that the method is in anyway practical, that is, that it can ever be used in practice; b) in which circumstances the method can be used; c) the most practical way of presenting the method.

With regard to c) the public relations project raised a number of issues. The logico-linguistic models and the empirical models are discrete logical events and constitute different stages in chapter 6. In practice it was not clear whether it was best to make these separate chronological stages in a real world project. This matter will be best resolved by conducting more real world project. A second issue raised by the project was the amount of model building input required from stake-holders and analysts. Clearly the early models should be stake-holder driven and the later conversion into Prolog analyst driven, but the intermediary stages are a gray area that needs to be resolved in practical projects.

The work can be seen in terms of the R & D model of industrial innovation. The main body of the thesis has provided the research. Development is a separate stage which began with the public relations project. It can be expected that many more projects will be needed before the method becomes a fully developed product. The classic SSM action research model, in which theory guides practice which in turn contributes to theory, is ideal for this type of work.

10.3 New directions for theory

10.3.1 Truth and programming

Philosophical logic has had two main concerns. On is to give a coherent explanation of how the concept of truth is used in natural languages. The other is to give a coherent theory of truth for artificial languages. The artificial language with which they have been almost exclusively concerned have been mathematics and logic. Formal systems of mathematics and logic are special cases in that they comprise very few rules from which an enormous number of theorems can be generated. A standard business computer system by contrast comprises enormous numbers of rules which can generate very few theorems.

Standard business computer systems often fail and the failure rate is of growing concern. When viewed from the perspective of philosophical logic it is surprising that they ever succeed. But the fact is that in most cases they do succeed; and this make the theory of truth that must be implicit in information system design interesting.

One of the most striking differences between information system designers and philosophers is the attitude towards stipulation. In the formulation of a formal logical or mathematical systems stipulation which is in terms of axioms, definitions and rules of production is kept to a minimum of self evident propositions. In the analysis of natural languages most philosophers of logic would recognize stipulation as in the case of naming. However, their attention is drawn not to what makes it true that an act of naming takes place, as when Adrian says *I shall call this dog "Rover"*, but to how it is true that an act of naming has taken place, as when Betty says *Adrian calls his dog "Rover"*.

Systems analysts, database designers and programmers make stipulations all the time. They stipulate, with hardly a thought, that every valid order has an order number and an order date. Admittedly if they are using SSADM these will not always be stipulations but representations of the existing bureaucracy. But this just pushes the problem one stage back - the bureaucrats made the stipulations. Also in green field information system design the computer people are forced to make the stipulations themselves.

The thesis has attempted to identify the logical mechanisms that can enable such a large amount of stipulation to work effectively in a real world situation. In order to do this the thesis has drawn heavily on theories of truth and meaning from philosophical logic. However, as the task has been quite different from that normally faced by a philosopher of logic an unusual account of truth has begun to emerge.

In brief, the implicit theory of truth in information system design and in programming is not coextensive with the theory of truth developed by philosopher to explain natural language or mathematics. A promising area for future research would, therefore, be to see if the ideas about truth and modality that have been used in this thesis can be developed into a theory of truth and modality that could cope equally well with the philosophers' concern about natural language and mathematical systems and with the implicit notions of truth contained in information system design and programming.

10.3.2 Theories of truth

Most theories concerning truth and meaning try to explain meaning in terms of truth. As was indicated in chapter 3 Wittgenstein held that "truth" was just another term in a language game. In other words he explained truth in terms of meaning. This idea has not been worked out in any great detail in philosophical logic.

This thesis has taken the language game theory as fundamental and has attempted to base a method of information system design upon it. Repeatedly the thesis has been drawn into arguments about the nature of truth. Although each of these arguments are not particularly unusual when they are drawn together and the rather surprising result is that a new theory of truth begins to emerge.

The fact that SSM models have such an unusual logical status plus the fact that we have been able to use rigorous arguments to develop a computerized system that can learn indicates that some light might have been thrown on the theory of truth. The formulation of a new theory of truth is not an undertaking that should be undertaken lightly but it does present an exciting idea for further research. The remainder of the this section will bring together arguments that have been touched upon in earlier chapters and draw up a tentative outline of what could be a new theory of truth.

Modal logic gives us two types of truth: factual truth and logical truth. One can be expressed in terms of the other:

$M\ p = -L\ -p$

$L\ q = -M\ -q$

From this it might seem that if we can establish a truth theory for one type of modal statement that we will have established a truth theory for the other.

The correspondence theory (which was discussed in chapter 3) works well for "*M*" statements so it might seem that the whole of modal logic can be explained in terms of the correspondence theory. However, there is a problem here. Although negative "*L*" statements can be derived from positive "*M*" statements (e.g. $M\ p \rightarrow -L\ -p$), positive "*L*" statements cannot be derived from a positive "*M*" statement or any set of positive "*M*" statements. This means that no matter how many positive facts we are able to establish on the basis of observations of the real world we will never be able to derive a positive "*L*" statement from them.

Nor can we derive a positive "*L*" statement from any set of negative statements about the real world. Suppose that "p" stands for "pigs can fly" the real world observation might lead us to assert "$M\ -p$" or possibly "$-M\ p$". But we require "$-M\ -p$" in order to derive "*L*", and it does not seem possible to support "$-M\ -p$" solely on the basis of observation. We cannot take "$-M\ -p$" to be a purely empirical statement, instead "$-M\ -p$" must be derived from "*L*". Thus we cannot, at least straight-forwardly, explain "*L*" statements in terms of the correspondence theory of truth.

A solution can be worked out as follows. We can accept the correspondence theory as a theory of reference. To say of an "*M*" statement that it is true is to say that it corresponds to a fact. We can now add an element from Tarski's semantic theory of truth and make the divide between object language and meta-language. An object language "R" will consist entirely of "*M*" statements. Saying that an "*M*" is true will be a statement in a meta-language "S". All "*L*" statements will also be part of the meta-language "S". "*L*" statements will be statements of the rules of "R". To say that an "*L*" statement is true will be to make a meta-meta-linguistic statement to the effect that "*L*" is a rule of "R". An "*L*" statement can only be true of "R" if it is not a contradictory of any other "*L*" statement, thus something of a coherence theory of truth can be used for "*L*" statements. "*L*" statements will describe the *sense* of the terms used in "R".

This combines elements of the correspondence, semantic and coherence theories of truth along with an explanation of the meaning of modal operators. It does not require Kripke's possible world semantics to explain modality but possible world semantics might follow from it. We now need to explain how "*L*" statements are established. This can be done using Wittgenstein's language game theory.

"*L*" statements are established by stipulation or agreement as in the case of SSM. They are something like edicts - statements of the law rather than statements about the law. A new "*L*" statement changes the language game, in fact it creates a new language and is, therefore, automatically true of that language.

If there is an existing language, say "RT", then we might wish to discover its rules and make statements about them. But all statements about "RT" will be contingent and will need to be discovered empirically. Suppose we think that "*a* -> *b*" is a rule of RT then we might express this as "*M L a* -> *b*". This means that we are making an inductive hypothesis to the effect that "*a* -> *b*" is logically true in RT. This in turn means that the people who speak RT have agreed to this rule. We can find out that "*a* -> *b*" is a rule of RT by observing the linguistic behavior of RT speakers.

10.3.3 Foundations for modal logic

The theory of truth given in the proceeding section suggests a system of modal logic which is somewhat different from existing systems. As was explained in section 8.1, in S5 the following is a theorem:

$$L\ (A) \rightarrow M\ (A)$$

If we read "L" as "by definition" and "M" as "contingently" then we have:

If is true by definition that all panthers are black then it is contingently true that all panthers are black.

This is obviously false. In fact the opposite is true as was demonstrated in 4.5.1. Given the theory of truth advocated here it would seem that "$L(A) \rightarrow -M(A)$" should be an axiom or theorem of the modal system. Further research is needed to determine if a modal system can be formulated that employs this as an axiom.

10.3.4 Limitations of Prolog

10.3.4.1 Theoretical limitations

One of the aims of the thesis has been to show how logico-linguistic modelling and the theoretical arguments behind them can work their way through into a computer program. Prolog was chosen mainly because it is easy for someone without a background in computing. The near English format makes it easy to see how statements and predicate logic can be expressed in a program.

However, previous chapters have shown that Prolog is limited in its ability to express formulas of the predicate calculus. There does not seem to be a way of making a Prolog program as powerful as a model in predicate logic.

The main limitation on Prolog is that although it is said to be a theorem prover (Forsyth, 1984, p. 13) it is not a theorem generator. It works on the chaining principle and this has more in common with a database and query language than with the predicate calculus. Essentially Prolog, like a database query language, works on the principle of search, find and list. The predicate calculus is quite different.

A system in the predicate calculus is made up of axioms and rules of production. One of the most simple rules of production is a rule of substitution concerning negation. This is that $-(-p)$ can be substituted for p. So, given the axiom p we can apply this rule and derive the theorem $-(-p)$ and this is written down as part of the system. Now we can apply the rule to $-(-p)$ and derive the theorem $-(-(-(-(p)$. This exercise in futility can repeated indefinitely and shows that this simple rule of production can generate an infinite number of theorems.

Given the axioms:

All Athenians are Greeks

All Greeks are mortal

the rule of production known as "Hypothetical Syllogism" will produce:

All Athenians are mortal

In the notation of the predicate calculus:

Fx: x is Athenian

Gx: x is Greek

Hx: x is a mortal

(Ax) Fx -> Gx

(Ax) Gx -> Hx

therefore,

(Ax) Fx -> Hx

We can proceed in this manner, generating as many theorems as we like, without worrying about whether there is an x or not. However, should we decide that there is an x, say socrates, then, by the production rule of Universal Introduction, we can substitute "socrates" throughout the system:

F(socrates) -> G(socrates)

G(socrates) -> H(socrates)

F(socrates) -> H(socrates)

In order to carry out this substitution you do not need to think about the nature of the logical connectives. Provided the axioms are not self contradictory any substitution will always produce a consistent system; this is because every formula in the system is either an axiom or a theorem of the system.

Prolog does not proceed in this way. Instead Prolog makes one substitution, say F(socrates), and then looks for others. It sees the rule F(socrates) -> Gx, and then makes another substitution i.e. G(socrates). In other words Prolog takes the implication sign in universal statements (those beginning with (Ax)) as being rules of production. Prolog is forced to behave this way because it allows contradictory statements to be written in the same program. That is there is no mechanism to prevent contradictory statements being entered into the system. These will compile and it is only when the program is run that the inconsistency may be detected.

Further research is needed to determine whether the type of Prolog program given at 6.3.5.2 or a similar Prolog program with the automatic deletion can have practical applications. Given that Prolog is one of the major artificial intelligence languages it is quite possible that it does. Further research is also needed to determine whether a more suitable program can be found to represent the modal logic in stage five of the six stage method. Lisp is a likely possibility.

10.3.4.2 Towards a true logic program

Prolog stands for "programming in logic" and it is now clear that this is something of a misnomer. It can be confusing because Prolog looks like logic and it can perform some of the functions of logic even though it does not function like logic. The shortcomings of Prolog for the expression of predicate logic gives us a good idea of how a proper logic program should perform.

Firstly, the program should perform in accordance with the rules of production for predicate logic, these would be built into the program. A particular program would start with a set of consistent universal statements which we can call "axioms". The rules of production would be allowed to operate on the axioms to produce theorems, these would all be universals. The variables could then be instantiated by rules of introduction. The resulting theorems would be particular statements. To find out if a particular statement is true the program would simply look it up. In this system chaining would be unnecessary.

New universals could be introduced provided they were consistent with the axioms. The new universals could be combined with the axioms and existing theorems to produced new theorems using the rules of production. A consistency checker should be easy to formulate as any statement that is not the negation of an axiom or a theorem will be consistent. This would be a proof theoretic system as apposed to a model theoretic system such as Prolog.

Another shortcoming of the Prolog programs that have been given in previous chapter is that although they can show that inductive hypotheses are false they cannot generate them. It seems that an algorithm capable of generating inductive hypotheses could be formulated in second order predicate logic. For example:

(Ex) (Ey) (Az) (AX) (AY) Xx & Yx & Xy & Xy & -(x = y) & -(Xz & -Yz)) -> (Yz ->Xz)

This means that if there are two objects x and y and the predicates X and Y apply to both objects, and there is no third object to which the predicate X applies but Y does not, then any object to which Y applies will be an object to which X applies.

Suppose that X is the property of being white and Y is the property of being a swan. in this case the formula means: if there are two swans that are both white and there is no swan that is not white then all swan are white.

A system incorporating this principle would employ classic induction. Unlike decision tree induction there would be no need to select and set the program up with a data set. Unlike neural networks it would not be confined to probabilities about a particular thing. It could function as a day to day expert system or even an order processing system and at the same time its knowledge would grow as more facts were fed into the system. Whether it is practical is a different matter as proof theoretic systems have tended to fall victim to the combinatorial explosion.

Such avenues of research are in any case closed to Prolog as it is confined to the first order predicate calculus (object variables only). It cannot handle the predicate variables that constitute the second order predicate calculus.

There is obviously a need for further research in this area. It needs to be determined whether Prolog is the best platform for the expression of logico-linguistic and empirical models. It is possible that another artificial intelligence language, such as LISP, might be better. Alternatively the development of a new platform written in a lower level language might be the best solution.

10.3.4.3 Carnap for structuring Prolog

An alternative platform is one solution to the limitations of Prolog but it might be found that other platform have similar if not worse limitations. Chapters 6 and 7 have shown that a workable Prolog program can be produced. Another direction for future research would be the further enhancement of the models prior to writing the Prolog.

Section 7.6.1 indicated that if it can be determined what information in terms of particular facts is likely to be available and which conclusions need to be reached, then this could provide guidelines for the choosing between the various ways in which the model can be expressed in Prolog.

Carnap and Bar-Hillel's theory of information might provide some theoretical foundation for this. A Prolog rule in which the antecedent is the consequent of few other Prolog rules and in which the consequent is the antecedent of many other Prolog rules will have a greater information content than a Prolog rule where the antecedent is the consequent of many other Prolog rules and in which the consequent is the antecedent of few other Prolog rules. For example:

```
Program A
dog (X) if canine (X).
clever (X) if dog (X).
can_be_trained (X) if clever (X).
make_good_guard (X) if clever (X).
```

```
Program B
dog (X) if likes_bones (X).
dog (X) if barks (X).
clever (X) if dog (X).
can_be_trained (X) if clever (X).
```

On something close to (but not precisely the same as) Carnap's theory we could say that the statement that dogs are clever has a greater information content in Program A than it does in Program B. Suppose we have one instantiation, say "Rover". In Program A it is only the fact that Rover is canine that would allow us to deduce that Rover is clever, while in Program B this can be deduced from the fact that Rover likes bones or from the fact that Rover barks. Given that Rover is clever Program A can deduce that Rover can be trained and that Rover will make a good guard but Program B will deduce only that Rover can be trained.

11 APPENDIXES

11.1 APPENDIX 1 - Logical notion

-p means not *p* (negation). It is true when *p* is false.

p & *q* means *p* and *q* (conjunction). *p & q* is true only if *p* and *q* are true.

p v *q* means *p* or *q* or both (alternation). *p v q* is true if *p* is true or if *q* is true or if *p* and *q* are true.

p -> *q* means if *p* then *q* (the conditional). *p -> q* is only false when *p* is true and *q* is false, otherwise it is true. Sometimes known as *implication*.

p <-> *q* means *p* if and only if *q* (the biconditional, sometimes known as logical equivalence or identity). *p <-> q* is true if *p* and *q* are both true or if *p* and *q* are both false, otherwise it is false.

(Ax) Fx "A" is used as the universal quantifier in the thesis. Where typesetting allows it is usually printed as an "A" upside down. It means "for all x". The formula here means that "F" is true of every x.

(Ex) Fx "E" is used as the universal quantifier in the thesis. Where typesetting allows it is usually printed as an "E" backwards. It means "there exist an x". The formula here means that "F" is true of at least one x.

11.2 APPENDIX 2 - Formal modal meta-rules

Section 6.3.4 contained the following modal meta-rules:

Meta-rule one: if A $\mid$- B then L(A) $\mid$- L(B)

Meta-rule two: if A $\mid$- B then M(A) $\mid$- M(B)

Meta-rule three: if A, B $\mid$- C then L(A), L(B) $\mid$- L(C)

Meta-rule four: if A, B $\mid$- C then M(A), L(B) $\mid$- M(C)

The first three meta-rules are intuitively obvious and will follow from the axiomatization of any system of modal logic. Meta-rule four seems to be obvious but its proof turns out to be quite complicated. The author had difficulty proving formula (15) in section 6.3.4. Following a discussion with the author, David Miller of the Philosophy Department, University of Warwick, came up with the idea of using meta-rules furnished the following proof of the following meta-rule:

if A, B $\mid$- C then L(A), M(B) $\mid$- M(C)

which is equivalent to Meta-rule four. Here is Miller's proof:

Although I need nothing like its strength, I'll use the axiomatization of S5 given in Chellas (1980, p. 14). In Chellas the box is used instead of "L" as the operator indicating necessity and the lozenge is used instead of "M" as the operator indicating contingency. The symbols "L" and "M" will continue to be used here. "¦-" is used for deducibility. (for deducibility in modal logic see Chellas, p. 47)

What I want to show is the validity of the meta-rule

if A, B ¦- C then L(A), M(B) ¦- M(C)

Since A, B ¦- C, we have by classical sentential [propositional] calculus that A -> (-C -> -B) is a tautology. Hence it is a theorem of almost any modal system.

By the rule RN of necessitation, then,

L(A -> (-C -> -B))

is a theorem. So by the rule K,

L(A -> (-C -> -B)) -> L(A) -> L(-C -> -B))

is a theorem. By MP (modus ponens),

L(A) -> L(-C -> -B)

is a theorem. By K again

$L(-C \to -B) \to (L(-C) \to L(-B))$

is a theorem, and by some routine (but boring) applications of PL and MP we may conclude that

$L(A) \to (L(-C) \to L(-B))$

is a theorem. But by definition M and some more sentential logic we see that $L(-C)$ is equivalent to $-M(C)$, and $L(-B)$ to $-M(B)$> Because logical equivalents are inter-substitutable, we conclude that

$L(A) \to (-M(C) \to -M(B))$

is a theorem, and so, at last,

$L(A) \to (M(B) \to M(C))$

is a theorem. By the definition of deducibility,

$L(A), M(B) \vdash M(C)$

Of course, quite a few steps have been left out here. As we know, axiomatic systems of modal logic are unwieldy to operate.

11.3 APPENDIX 3 - Diagrammatic Techniques

11.3.1 Compiling the dictionary

11.3.1.1 Introduction

The purpose of this dictionary is to lay out the logico-linguistic modelling diagramming techniques along with the equivalent statements in propositional logic, predicate logic and Prolog.

The logico-linguistic diagraming convention here differs from those used in some previously published papers (see Declaration for list of papers). The diagramming technique evolved slowly and progressed as research indicated the need to include more and more logical relations. In the first paper, "Causation & Soft Systems Models", a broken arrow had been added to the traditional SSM diagramming vocabulary. By the time "Mapping Conceptual Models on to the Real World" was written the models included broken arrows, double headed arrows, containing bubbles and model flags. All of these had been added on an *ad hoc* basis.

The general flow of the arrows used in the Checkland models has been preserved because this matched Prolog fairly well. For example the Checkland statement "Obtain raw material" with an arrow to "Make product" would correspond to the

Prolog statement "raw_material_obtained (X) :- product_made (X)." Both statements are saying that if the product has been made then the raw materials must have been obtained.

It is most confusing that the direction of the arrow of implication is the opposite way round in both propositional and predicate logic. In logic the statement would be "product made -> raw material obtained". For many readers it will be easier to go from the bubble diagrams directly to Prolog and the diagramming technique has been constructed accordingly. The logic is still important because it provides the justification for the translations. In natural English the arrow of implication can go both ways: "Raw materials will have been obtained if the product has been made"; "If the product has been made then raw materials will have been obtained"; "The product can be made only if the raw materials are obtained".

The compilation of the dictionary was not in itself intended to be a work of any academic importance. However, there were some findings that may be of some interest.

11.3.1.2 The superfluous broken line

If we equate q being a necessary condition of p with $-(p \& -q)$

or p -> q; and if we equate r as a sufficient condition of s then r -> s and so we can say that if q is necessary for p then p is sufficient for q. This holds in all cases even where the necessary or sufficient conditions are compounds of & or v. Thus the broken arrow is not logically required - the same statement can be made by reversing the arrows.

However, there are reason for including the broken arrow. Psychologically if we are looking for the cause of p it seems strange to require that it be pointed out that p is a necessary condition of r. However, if p is a necessary condition of r then r is a sufficient condition of p and therefore relevant to a causal account of p. Psychologically it is much better to describe r as a sufficient condition.

Sufficiency and necessity are not exactly the same as implication. To say that q is part of the cause of p is to say that q did not happen later than p. There is an implicit temporal reference in the relations of necessity and sufficiency. This can only be eliminated if temporal references are brought out. Thus p is necessary for q should be rendered as:

q -> p and q is an event not later than p.

The arrows will only function as pure logical connectives if the temporal references are included in the bubbles. This has been done in the figure 7. Here every case of a broken arrow can be replaced by a solid arrow in the opposite direction and by the same token every solid arrow can be replaced by a broken arrow in the opposite direction.

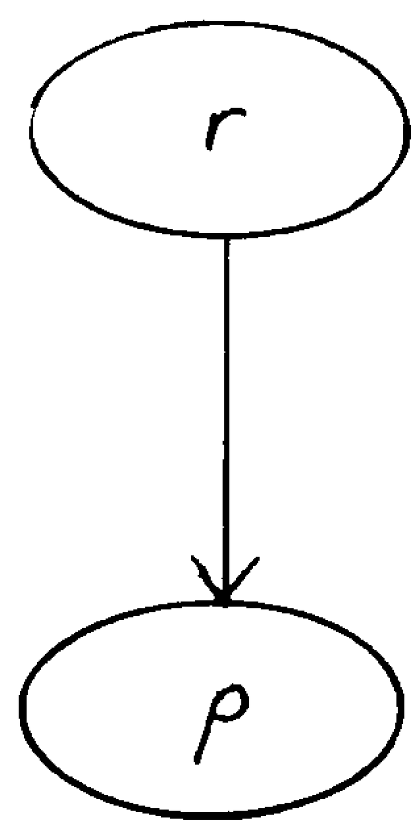
Time references in English make it very difficult to think without the dotted arrow, as the author found when he tried to build models that left it out. Losing the dotted arrow does not make model construction easier. The same is true in logical notation of the four logical connectives "&", "v", "->", "<->" two are superfluous and can be expressed in terms of the other two. They are retained only because psychologically they make it easier to render English into logic.

However, if might be found easier to express the models in Prolog if the broken arrows are replaced with solid arrows prior to writing the programs.

11.3.2 Dictionary - diagrams, logic & Prolog

11.3.2.1 Single solid arrow

Logic-linguistic diagram, type 1



Examples:

Ruddles is sold if people are happy.

If people are happy then Ruddles is sold.

People are happy only if Ruddles is sold.

Propositional Calculus:

$p \rightarrow r$

Predicate Calculus:

Px: x is a situation in which people are happy

Rx: x is a situation in which Ruddles is sold

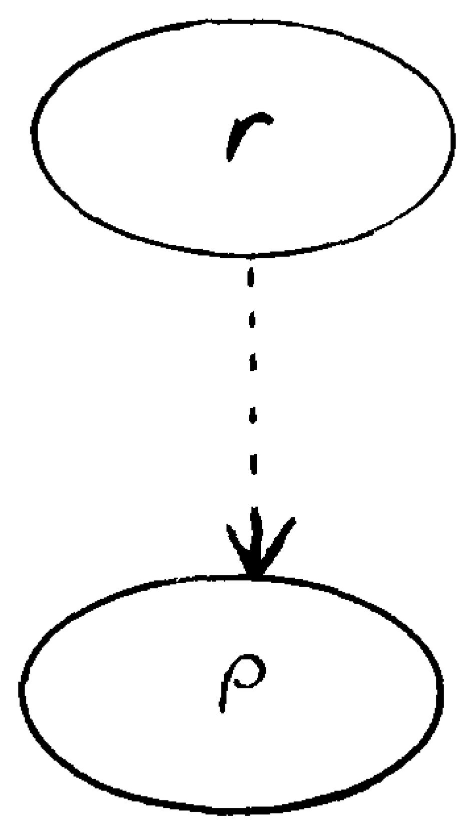
(Ax) Px -> Rx

Prolog:

```
ruddles_sold (X) :-
     people_happy (X).
```

11.3.2.2 Single broken arrow

Logic-linguistic diagram, type 2



Examples:

People are happy if Ruddles is sold.

If Ruddles is sold then people are happy.

Ruddles is sold only if people are happy.

Propositional Calculus:

$r \rightarrow p$

Predicate Calculus:

Px: x is a situation in which people are happy

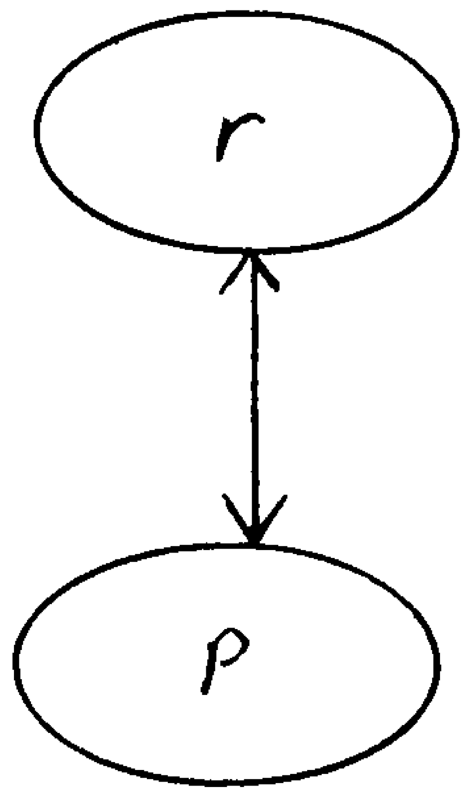
Rx: x is a situation in which Ruddles is sold

(Ax) Rx -> Px

Comment: The broken arrow is a useful device in knowledge elicitation but it is logically redundant, it is the logical equivalent of a solid arrow pointing in the opposite direction. Prior to converting the model into Prolog it is a good idea to replace all broken arrows with solid arrows. When this is done we see that the solid arrow goes the opposite way to that in the previous example. The combination of the two examples results in a double headed arrow and is the same thing as the next example.

11.3.2.3 Double headed arrow

Logic-linguistic diagram, type 3



Example: Ruddles is sold if people are happy and people are happy if Ruddles is sold.

Propositional Calculus:

r <-> *p*

Predicate Calculus:

(Ax) Rx <-> Px

This the same as the conjunction of:

(Ax) Rx -> Px

and

(Ax) Px -> Rx

Prolog:

```
ruddles_sold (X) :-
     people_happy (X).
not_people_happy (X) :-
      not_ruddles_sold (X).
```

Explanation: We cannot express this as "ruddles_sold (X) :- people_happy (X). people_happy (X) :- ruddles_sold (X)." because this would introduce the loop. So we express one half of the biconditional using the "not" predicates. In causal terms type 1 arrows show r as a necessary condition of p, type 2 arrows show r as a sufficient condition of p, and type three arrows are a conjunction of the two i.e. a necessary and sufficient condition.

This will enable us to make all the deductions that follow from the double headed arrow. If we know whether people are happy or not we will be able to work out if ruddles is sold and if we know whether people are happy or not. Suppose we know that people are happy at the Bull, that Ruddles is sold at the Bear, people are not happy at the Stag and Ruddles is not sold at the Stork. Using the double entry data system we enter:

```
ruddles_sold (the_bear).
not_ruddles_sold (not_the_bear).
ruddles_sold (not_the_stork).
```

```
not_ruddles_sold (the_stork).
people_happy (the_bull).
not_people_happy (not the_bull).
people_happy (not_the_stag).
not_people_happy (the_stag).
```

Given this Prolog returns the following:

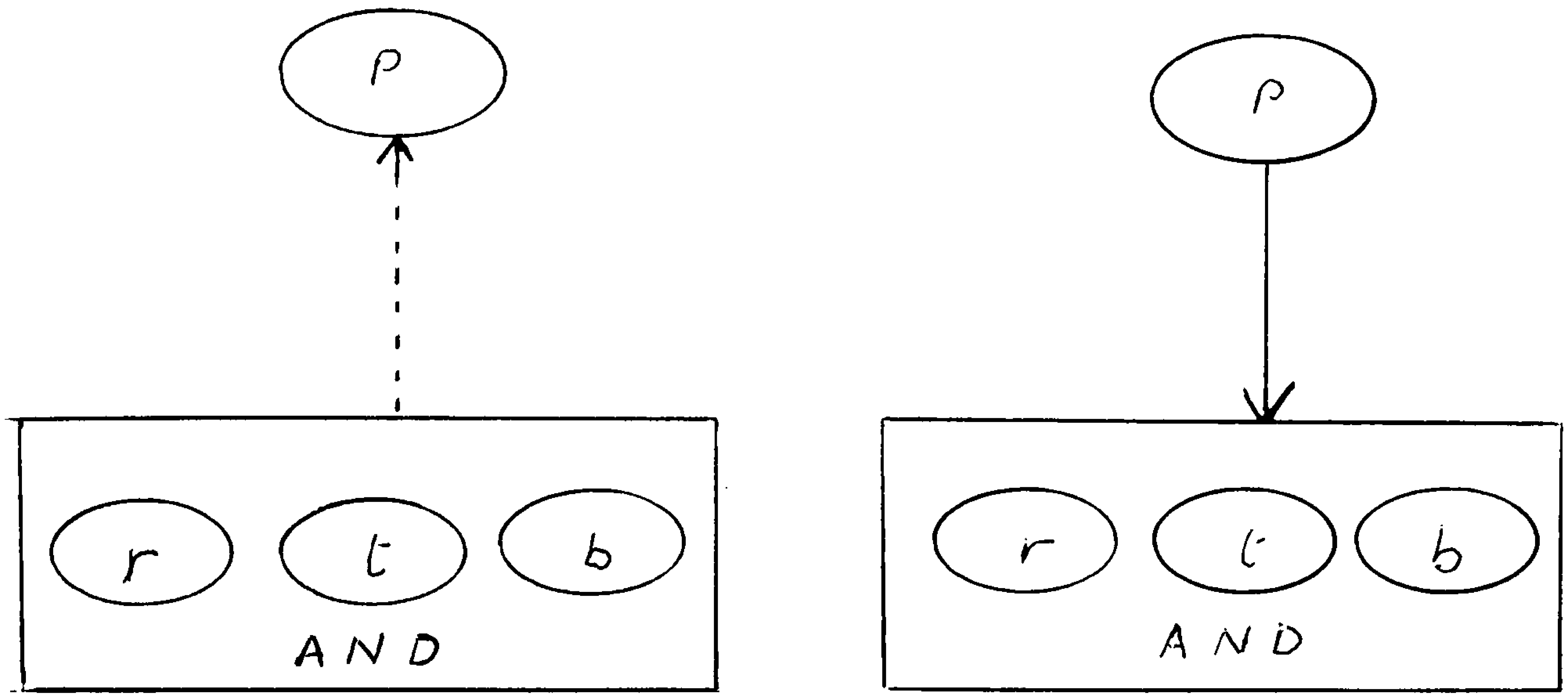
```
Goal: ruddles_sold (X)
X = the_bull
X = not_the_stag
X = the_bear
X = not_the_stork
```

```
Goal: not_people_happy (X)
X = not_the_bear
X = the_stork
X = not_the_bull
X = the_stag
```

In other words people are happy at the Bear and at the Bull.

11.3.2.4 AND, solid arrow leading in

Logic-linguistic diagram, type 4



The solid arrow on the right-hand diagram can replace the redundant broken arrow in the left-hand diagram.

Example: People are happy if Ruddles, Theakstons and Boddingtons are sold.

If Ruddles, Theakstons and Boddingtons are sold then people are happy.

Propositional Calculus:

$(r \ \& \ t \ \& \ b) \rightarrow p$

Predicate Calculus:

(Ax) ((Rx & Tx & Bx) -> Px)

This is a horn clause.

Prolog:

Alternative A

```
people_happy (X) :-
     ruddles_sold (X),
     theakstons_sold (X),
     boddingtons_sold (X).
```

Alternative B

```
not_ruddles_sold (X) :-
     not_people_happy (X),
     theakstons_sold (X),
     boddingtons_sold (X).
not_theakstons_sold (X) :-
     not_people_happy (X),
     boddingtons_sold (X),
     ruddles_sold (X).
not_boddingtons_sold (X) :-
     not_people_happy (X),
     theakstons_sold (X),
     ruddles_sold (X).
```

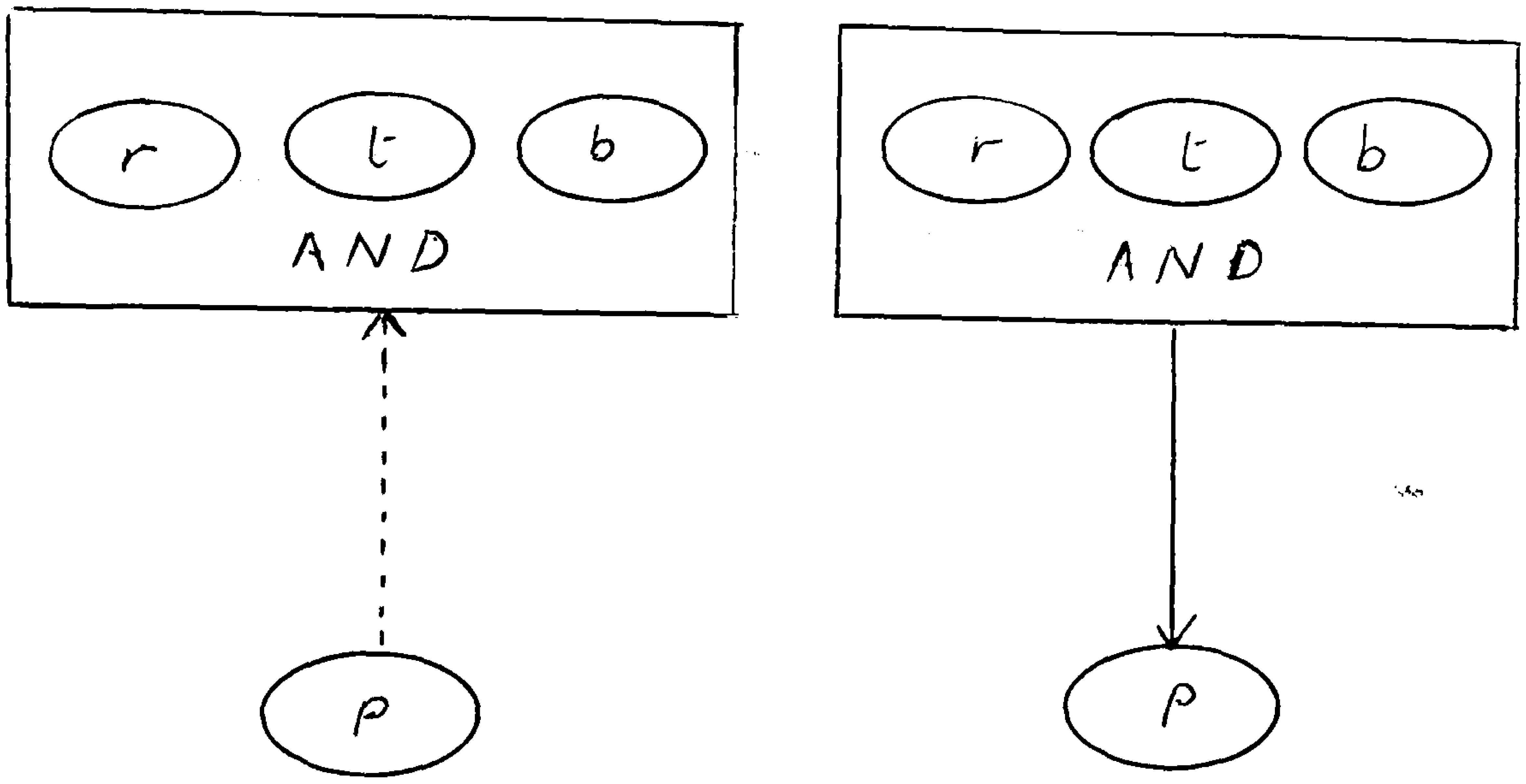
Explanation: These two alternatives cannot both be part of a program without circularity.

Alternative A maximizes the power of inferences in the direction of "*p*". Information about beer sales will allow inferences about people being happy. This program will infer that people are happy in those cases where all three beers are sold. It cannot make any inferences about which beers are sold.

Alternative B maximizes the power of inferences in the direction of "*r*, *t*, *b*". Instances where all three beers are sold must be instances where people are happy. Therefore, if there is a case where people are not happy and the two types of beer are sold then it is not possible for the other type of beer to be sold. This, however, will be able to make any inference about people being happy.

11.3.2.5 AND, solid arrow leading out

Logic-linguistic diagram, type 5



Again the broken line is redundant and can be replaced with the solid line going in the opposite direction.

Examples: Ruddles, Theakstons and Boddingtons are sold if people are happy.

If people are happy then Ruddles, Theakstons and Boddingtons are sold.

It will not be true that people are happy unless it is true that Ruddles, Theakstons and Boddingtons are sold.

Propositional Calculus:

$p \rightarrow (r \,\&\, t \,\&\, b)$

Predicate Calculus:

Tx: x is a situation in which Theakstons is sold

Bx: x is a situation in which Boddingtons is sold

(Ax) Px -> (Rx & Tx & Bx)

Prolog:

Alternative A

```
not_people_happy (X) :-
     not_ruddles_sold (X);
     not_theakstons_sold (X);
     not_boddingtons_sold (X).
```

Alternative B

```
ruddles_sold (X) :-
     people_happy (X).
theakstons_sold (X) :-
     people_happy (X).
boddingtons_sold (X) :-
     people_happy (X).
```

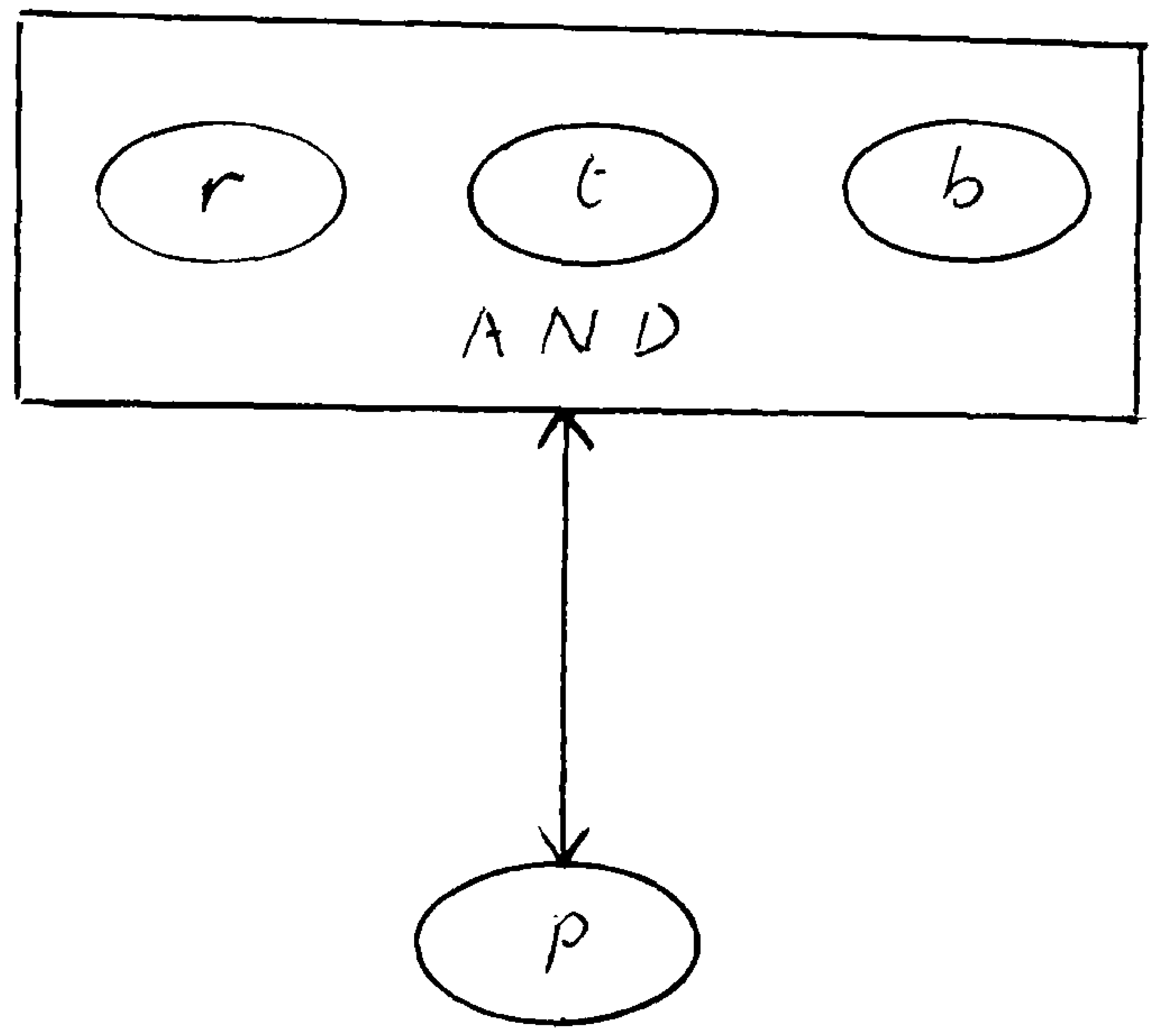
Explanation: These two alternatives cannot both be part of a program without circularity.

Alternative A maximizes inferences about whether people are unhappy. The sales of all three types of beer are a necessary condition of people being happy, therefore, any instance of one of the beers not being sold will be an instance of people being unhappy.

Alternative A maximizes inferences about which beers are sold. Being happy is a sufficient condition of all three beers being sold, therefore, if people are happy each of the beers must be sold.

11.3.2.6 AND, double headed arrow

Logic-linguistic diagram, type 6



Examples: Ruddles, Theakstons and Boddingtons are sold if people are happy and people are happy if Ruddles, Theakstons and Boddingtons are sold.

Propositional Calculus:

p <-> (r & t & b)

Predicate Calculus:

Tx: x is a situation in which Theakstons is sold

Bx: x is a situation in which Boddingtons is sold

(Ax) Px <-> (Rx & Tx & Bx)

Prolog:

Alternative A

```
people_happy (X) :-
     ruddles_sold (X),
     theakstons_sold (X),
     boddingtons_sold (X).
not_people_happy (X) :-
     not_ruddles_sold (X);
     not_theakstons_sold (X);
     not_boddingtons_sold (X).
```

Alternative B

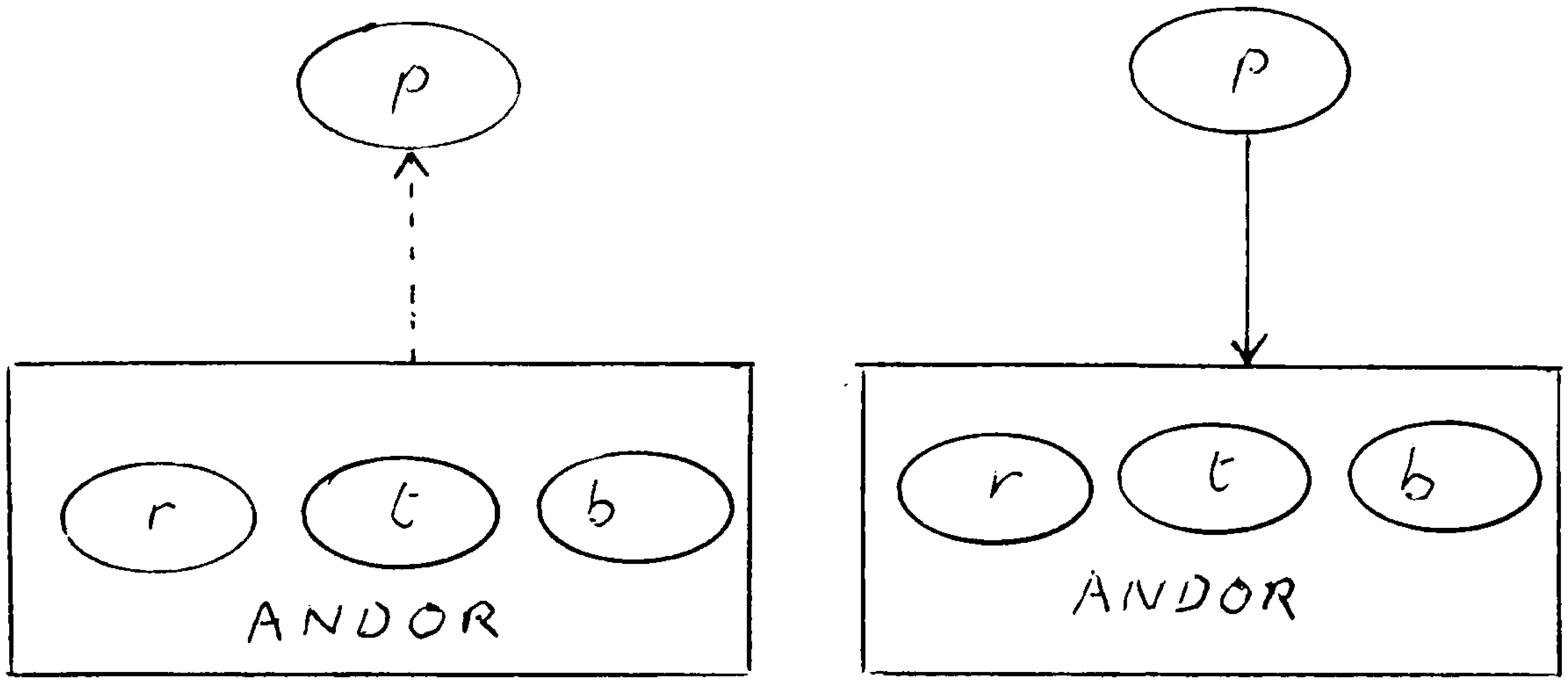
```
not_ruddles_sold (X) :-
     not_people_happy (X),
     theakstons_sold (X),
     boddingtons_sold (X).
not_theakstons_sold (X) :-
     not_people_happy (X),
     boddingtons_sold (X),
     ruddles_sold (X).
not_boddingtons_sold (X) :-
     not_people_happy (X),
     theakstons_sold (X),
     ruddles_sold (X).
ruddles_sold (X) :-
     people_happy (X).
```

```
theakstons_sold (X) :-
     people_happy (X).
boddingtons_sold (X) :-
     people_happy (X).
```

Explanation: The double headed arrow is the biconditional and is the combination of types 4 and 5. It can be constructed by combining the alternative A from 4 and 5 or by combining the alternatives B from 4 and five. The former maximizes inferences in the direction of people being happy or not, the latter inferences in the direction of what beers are sold.

11.3.2.7 ANDOR, arrow leading in

Logic-linguistic diagram, type 7



Example: People are happy if one or more of Ruddles, Theakstons or Boddingtons is sold.

If any of Ruddles, Theakstons or Boddingtons is sold then people are happy.

Propositional Calculus:

$(r \vee t \vee b) \rightarrow p$

Predicate Calculus:

(Ax) (Rx v Tx v Bx) -> Px

Prolog:

Alternative A

```
people_happy (X) :-
     ruddles_sold (X);
     theakstons_sold (X);
     boddingtons_sold (X).
```

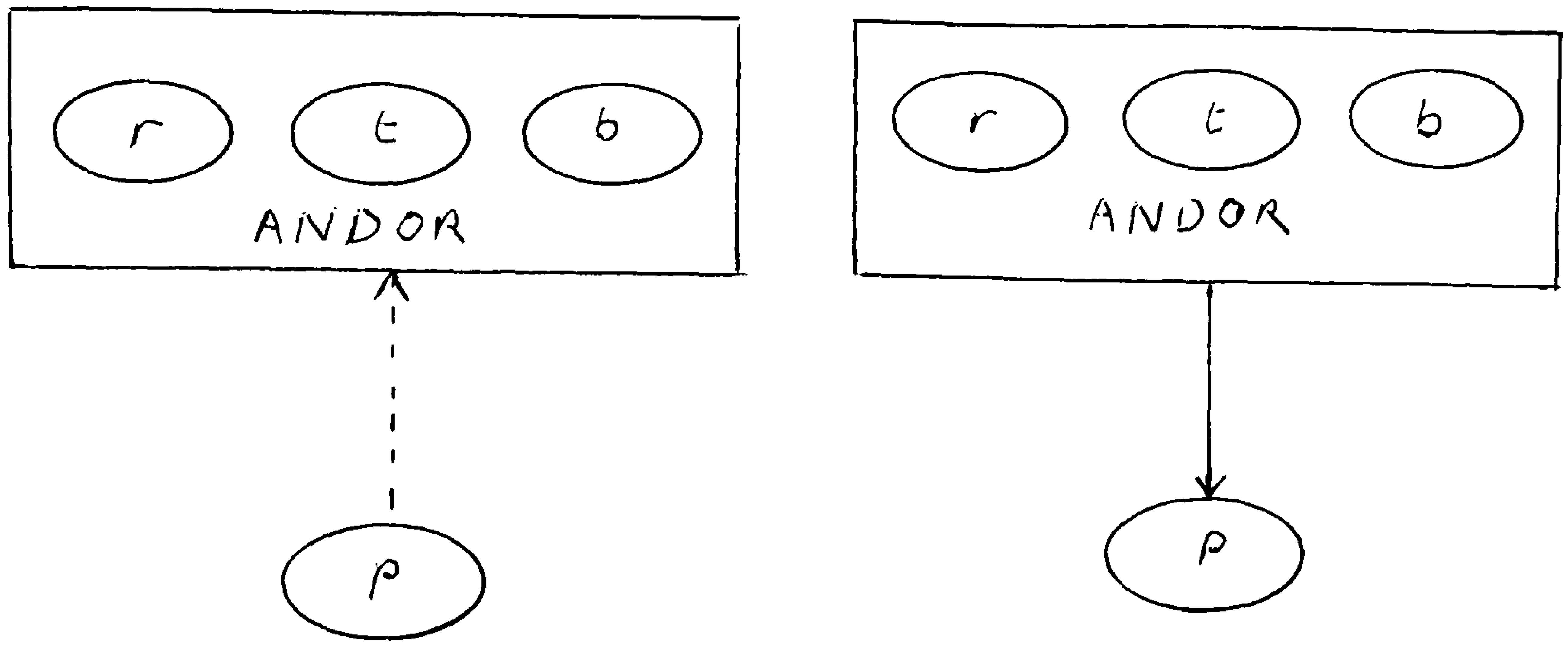
Alternative B

```
not ruddles_sold (X) :-
     not_people_happy (X).
not_theakstons_sold (X) :-
     not_people_happy (X).
not_boddingtons_sold (X) :-
     not_people_happy (X).
```

Explanation: Alternative a. follows from the fact that either one of the beers being sold is sufficient for people to be happy. Alternative b. from the fact that at least one of the beers being sold is dependent upon people being happy therefore if people are not happy then none of the beers can have been sold.

11.3.2.8 ANDOR, arrow leading out

Logic-linguistic diagram, type 8



Examples: One or more of Ruddles, Theakstons or Boddingtons is sold if people are happy.

If people are happy then one or more of Ruddles, Theakstons or Boddingtons is sold.

Propositional Calculus:

p -> (*r* v *t* v *b*)

Predicate Calculus:

(Ax) Px -> (Rx v Tx v Bx)

Prolog:

Alternative A

```
not_people_happy (X) :-
     not_ruddles_sold (X),
     not_theakstons_sold (X)),
     not_boddingtons_sold (X)).
```

Alternative B

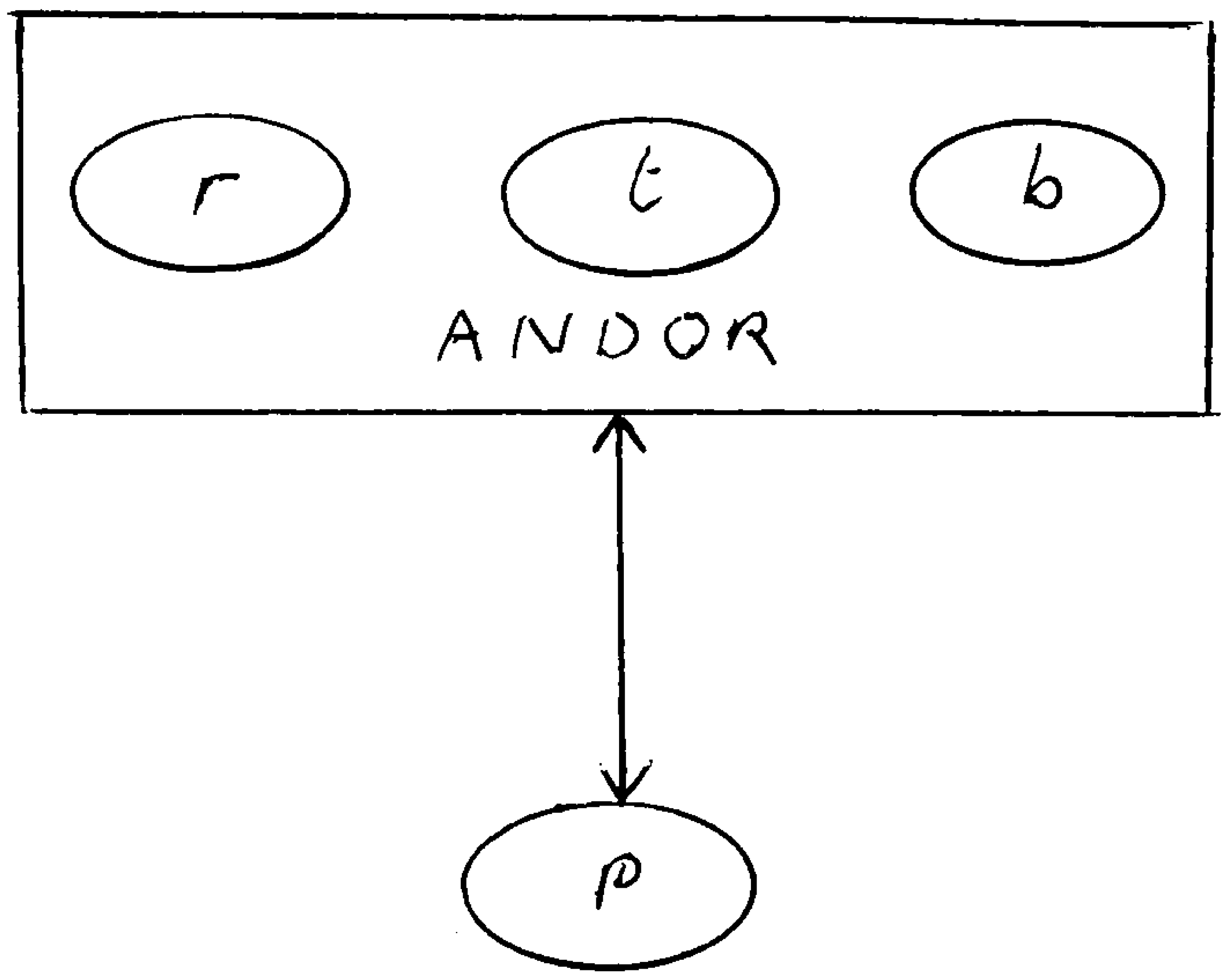
```
ruddles_sold (X) :-
     people_happy (X),
     not_theakstons_sold (X),
     not_boddingtons_sold (X).
theakstons_sold (X) :-
     people_happy (X),
     not_ruddles_sold (X),
     not_boddingtons_sold (X).
boddingtons_sold (X) :-
     people_happy (X),
     not_theakstons_sold (X),
     not_ruddles_sold (X).
```

Explanation: Alternative A, people being happy is sufficient for at least one of the beers to be sold. Therefore, if none of the beers are sold them people cannot be happy.

Alternative B, at least one of the beers must be sold if people are happy. So, given that people are happy and two of the beers are not sold the other beer must be sold.

11.3.2.9 ANDOR, double headed arrow

Logic-linguistic diagram, type 9



Example: One or more of Ruddles, Theakstons or Boddingtons is sold if people are happy and people are happy if one or more of Ruddles, Theakstons or Boddingtons is sold.

Propositional Calculus:

$p \leftrightarrow (r \vee t \vee b)$

Predicate Calculus:

(Ax) Px <-> (Rx v Tx v Bx)

Prolog:

Alternative A

```
people_happy (X) :-
     ruddles_sold (X);
     theakstons_sold (X);
     boddingtons_sold (X).
not_people_happy (X) :-
     not_ruddles_sold (X),
     not_theakstons_sold (X)),
     not_boddingtons_sold (X)).
```

Alternative B

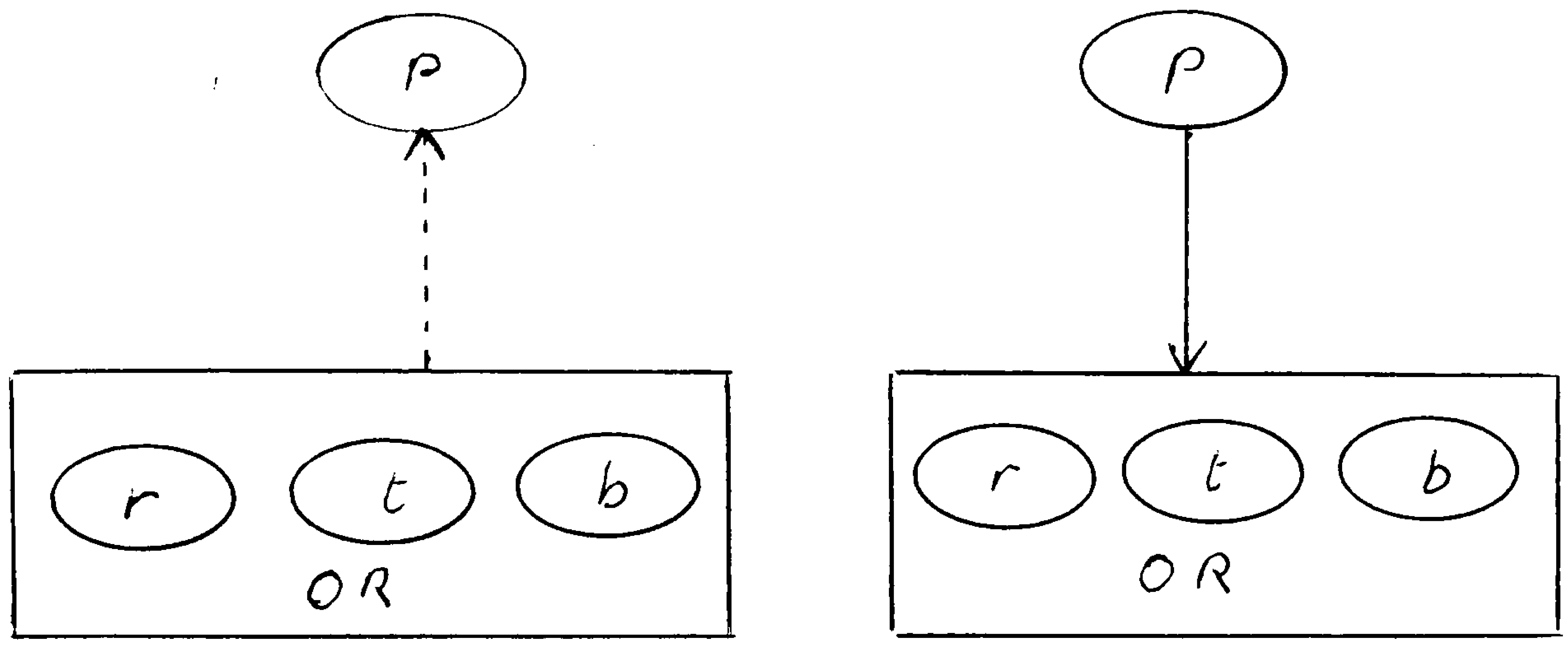
```
not_ruddles_sold (X) :-
     not_people_happy (X).
not_theakstons_sold (X) :-
     not_people_happy (X).
not_boddingtons_sold (X) :-
     not_people_happy (X).
ruddles_sold (X) :-
     people_happy (X),
     not_theakstons_sold (X),
     not_boddingtons_sold (X).
theakstons_sold (X) :-
     people_happy (X),
     not_ruddles_sold (X),
     not_boddingtons_sold (X).
boddingtons_sold (X) :-
```

```
people_happy (X),
not_theakstons_sold (X),
not ruddles_sold (X).
```

Explanation: Alternative A is the conjunction of the type the type 7 and type 8 alternative A. Alternative B is the conjunction of the type the type 7 and type 8 alternative B.

11.3.2.10 OR, arrow leading in

Logic-linguistic diagram, type 10



Example: People are happy if only one of Ruddles, Theakstons or Boddingtons is sold.

Propositional Calculus:

$((r \mathbin{\&} -(t \vee b)) \vee (t \mathbin{\&} -(r \vee b)) \vee (b \mathbin{\&} -(r \vee t))) \to p$

Predicate Calculus:

(Ax) ((Rx & -Tx & -Bx) v (Tx & -Rx & -Bx) v (Bx & -Rx & -Tx)) -> Px

This is a horn clause.

Prolog:

Alternative A

```
people_happy (X) :-
     ruddles_sold (X),
     not_theakstons_sold (X),
     not_boddingtons_sold (X);
     theakstons_sold (X),
     not_boddingtons_sold (X),
     not_ruddles_sold (X);
     boddingtons_sold (X),
     not_ruddles_sold (X),
     not_theakstons_sold (X).
```

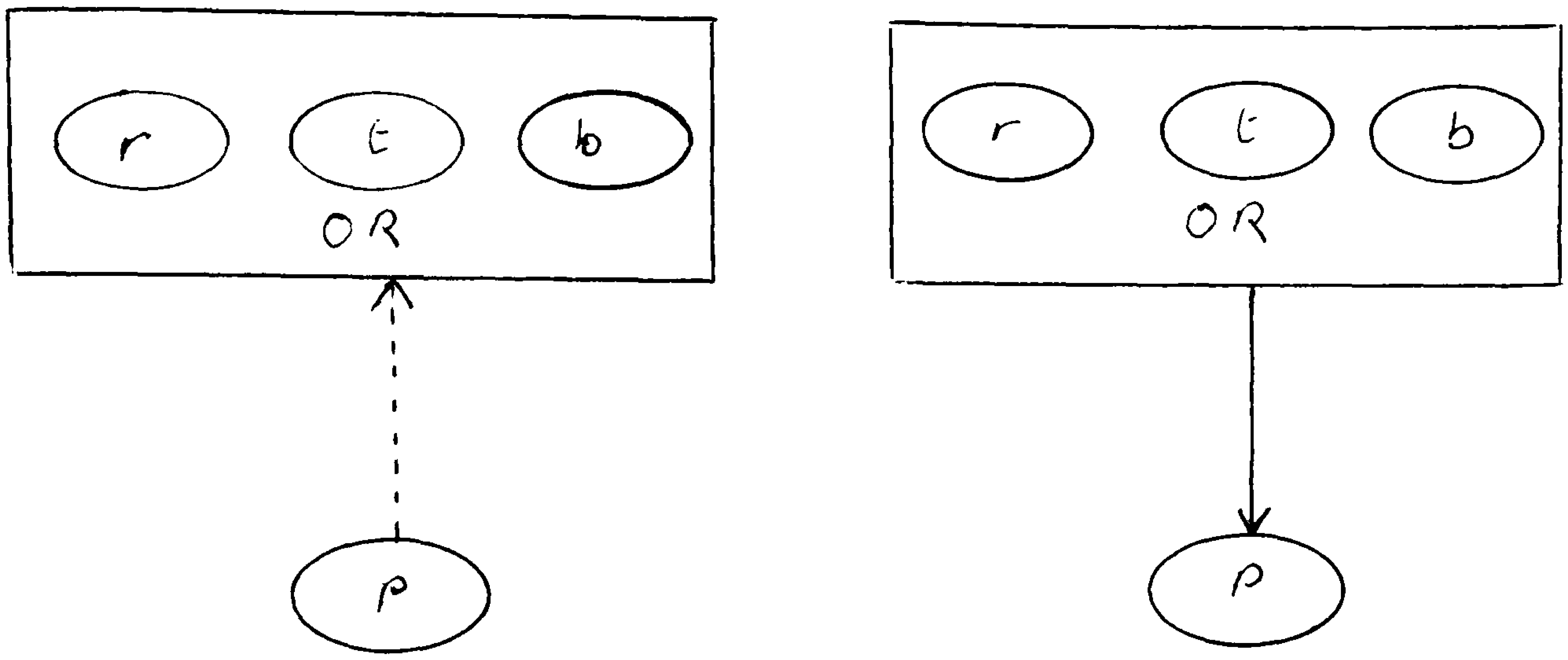
Alternative B

```
not_ruddles_sold (X) :-
     not_people_happy (X).
not_theakstons_sold (X) :-
     not_people_happy (X).
not_boddingtons_sold (X) :-
     not_people_happy (X).
```

Explanation: OR represents exclusive alternation which means that one of the beers being sold but not more than one of the beers being sold is a sufficient condition of people being happy. Alternative A follows from the fact that if any one beer is sold and the other two are not then people will be happy. Alternative B follows from the fact that people are happy is a necessary condition of any beer being sold, therefore, if people are not happy none of the beers are sold.

11.3.2.11 OR, arrow leading out

Logic-linguistic diagram, type 11



Example: One and only one of Ruddles, Theakstons or Boddingtons is sold if people are happy.

If people are happy then one and only one of Ruddles, Theakstons or Boddingtons is sold.

Propositional Calculus:

p -> ((r & -(t v b)) v (t & -(r v b)) v (b & -(r v t)))

Predicate Calculus:

(Ax) Px -> ((Rx & -(Tx v Bx)) v (Tx & -(Rx v Bx)) v (Bx & -(Rx v Tx))

Prolog

Alternative A

```
not_people_happy (X) :-
     not_ruddles_sold (X);
     not_theakstons_sold (X);
     not_boddingtons_sold (X).
```

Alternative B

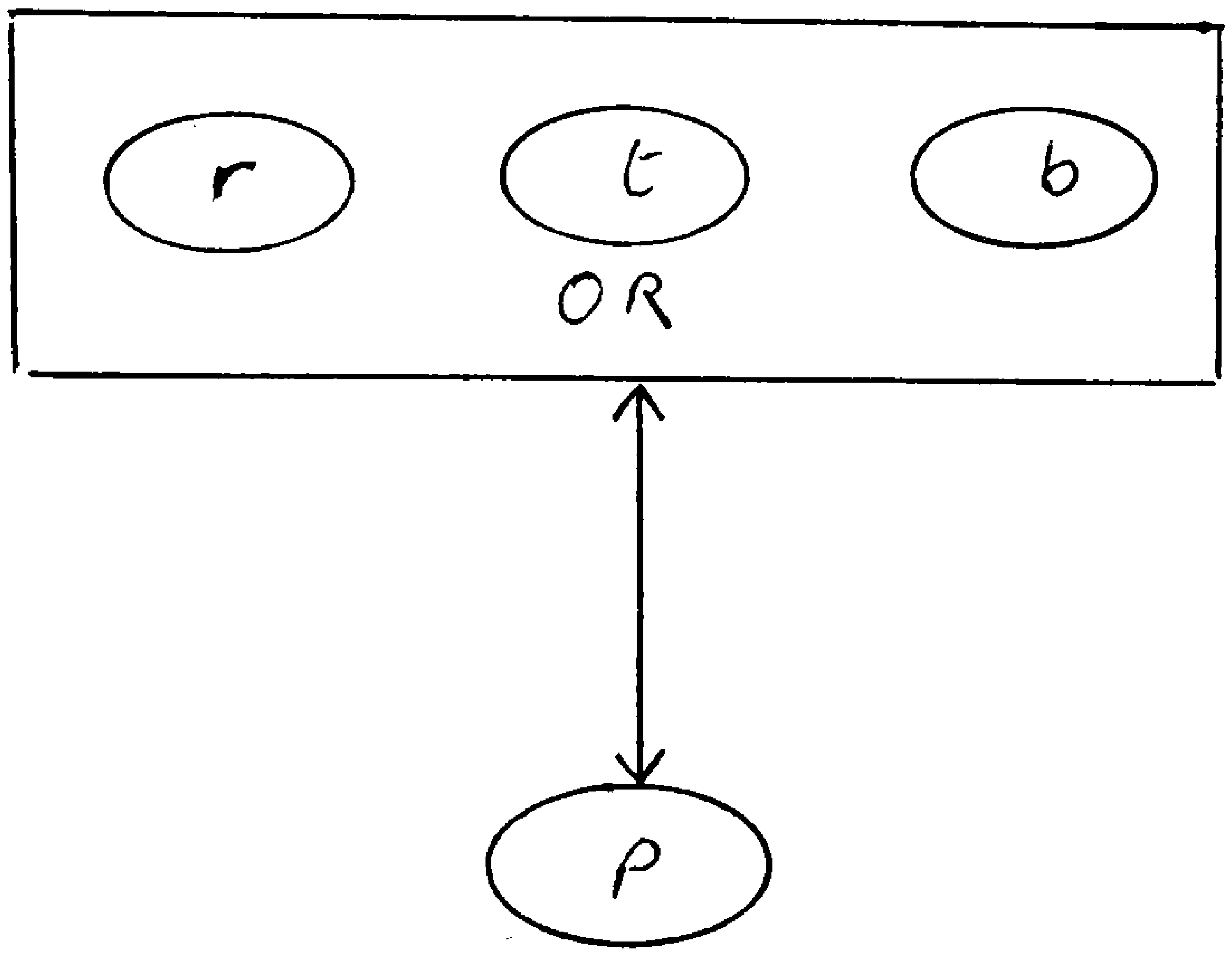
```
not_ruddles_sold (X) :-
     people_happy (X),
     theakstons_sold (X);
     boddingtons_sold (X).
not_theakstons_sold (X) :-
     people_happy (X),
     ruddles_sold (X);
     boddingtons_sold (X).
not_boddingtons_sold (X) :-
     people_happy (X),
     theakstons_sold (X);
     ruddles_sold (X).
```

Explanation: Alternative A follows from the fact that at least one of the beers being sold is a necessary condition of people being happy. Therefore, if none of the beers are sold people cannot be happy. Alternative B is more complex. The Prolog program give above as type 8 Alternative B (the one that begins "ruddles_sold (X) :- people_happy (X), not_theakstons_sold (X), not_boddingtons_sold (X).") also follows from the propositional and predicate formulas given here. The formula given here, however, is more powerful because less information is required to make a deduction. In order to deduce that Ruddles is sold we need to know that Theakstons is not sold *and* that Boddingtons is not sold. The program given here allows the deduction that Ruddles is not sold from the fact that Theakstons is not sold *or* from the fact that Boddingtons is not sold, therefore we do not need to know as much.

There are, therefore, at least three ways this can be expressed in Prolog.

11.3.2.12 OR, double headed arrow

Logic-linguistic diagram, type 12



Example: Only one of ruddles, theakstons or boddingtons is sold if people are happy and people are happy if only one of Ruddles, Theakstons or Boddingtons is sold.

Propositional Calculus:

p <-> ((*r* & -(*t* v *b*)) v (*t* & -(*r* v *b*)) v (*b* & -(*r* v *t*)))

Predicate Calculus:

(Ax) Px <-> ((Rx & -(Tx v Bx)) v (Tx & -(Rx v Bx)) v (Bx & -(Rx v Tx))

Prolog:

Alternative A

```
people_happy (X) :-
     ruddles_sold (X),
     not_theakstons_sold (X),
     not_boddingtons_sold (X);
     theakstons_sold (X),
     not_boddingtons_sold (X),
     not_ruddles_sold (X);
     boddingtons_sold (X),
     not_ruddles_sold (X),
     not_theakstons_sold (X).
not_people_happy (X) :-
     not_ruddles_sold (X);
     not_theakstons_sold (X);
     not_boddingtons_sold (X).
```

Alternative B

```
not_ruddles_sold (X) :-
     not_people_happy (X).
not_theakstons_sold (X) :-
     not_people_happy (X).
not_boddingtons_sold (X) :-
     not_people_happy (X).
not_ruddles_sold (X) :-
     people_happy (X),
     theakstons_sold (X);
     boddingtons_sold (X).
```

```
not_theakstons_sold (X) :-
     people_happy (X),
     ruddles_sold (X);
     boddingtons_sold (X).
not_boddingtons_sold (X) :-
     people_happy (X),
     theakstons_sold (X);
     ruddles_sold (X).
```

Explanation: These are just the combination of types 10 and 11.

Figure 1

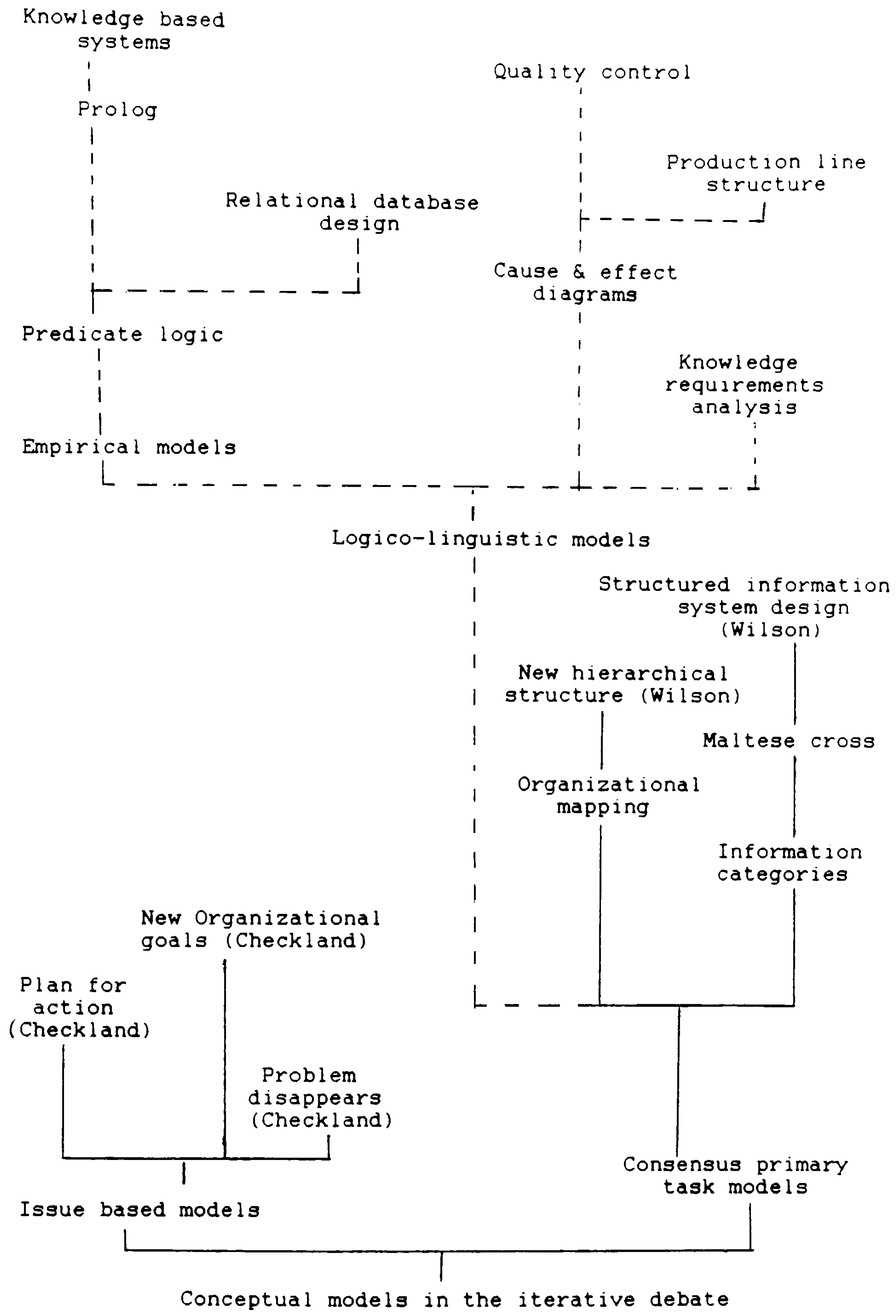

Knowledge based systems
Prolog
Relational database design
Predicate logic
Empirical models
Quality control
Production line structure
Cause & effect diagrams
Knowledge requirements analysis
Logico-linguistic models
Structured information system design (Wilson)
New hierarchical structure (Wilson)
Maltese cross
Organizational mapping
Information categories
New Organizational goals (Checkland)
Plan for action (Checkland)
Problem disappears (Checkland)
Consensus primary task models
Issue based models
Conceptual models in the iterative debate

Figure 2

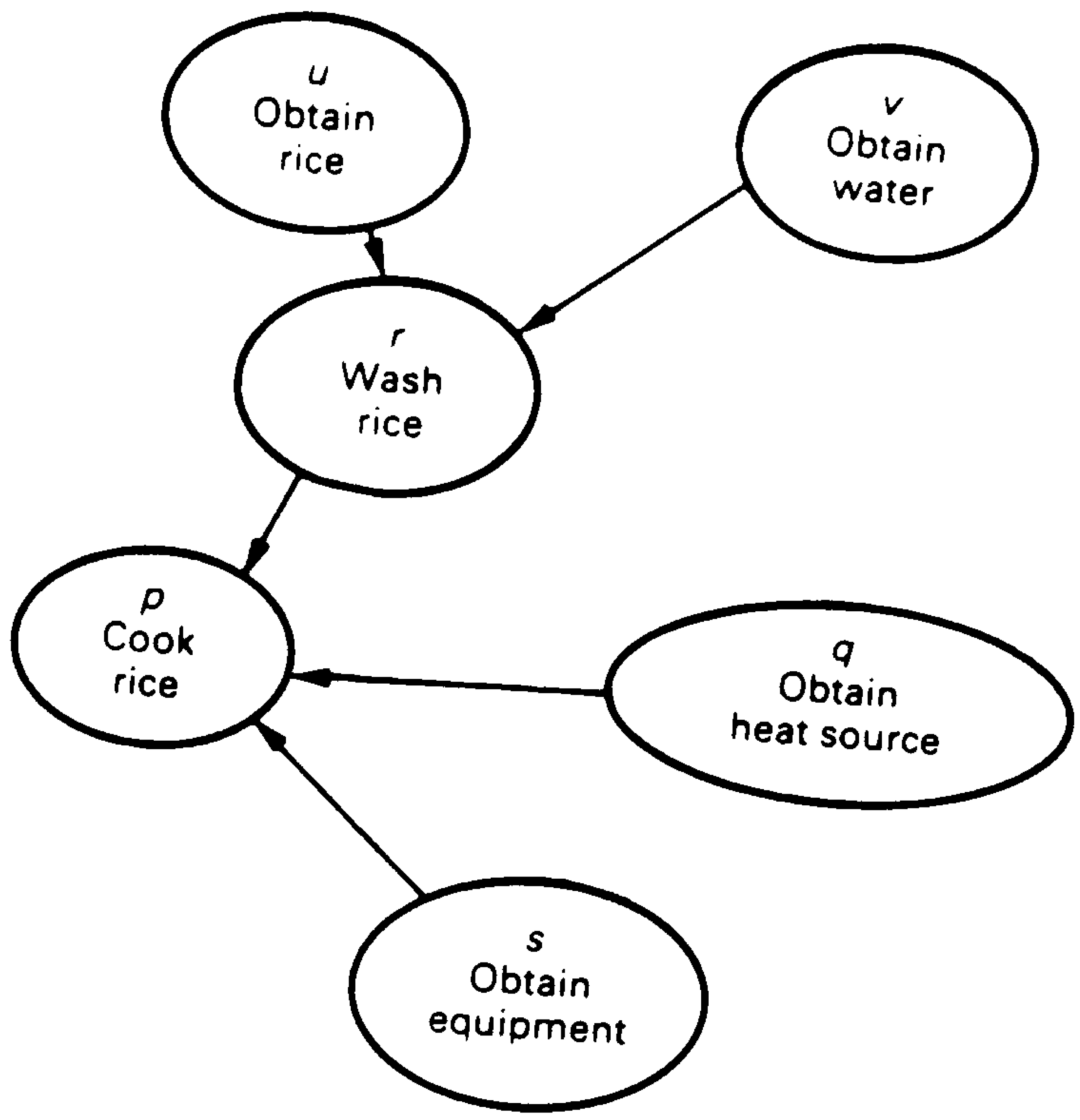

u
Obtain
rice
v
Obtain
water
r
Wash
rice
p
Cook
rice
q
Obtain
heat source
s
Obtain
equipment

Figure 3

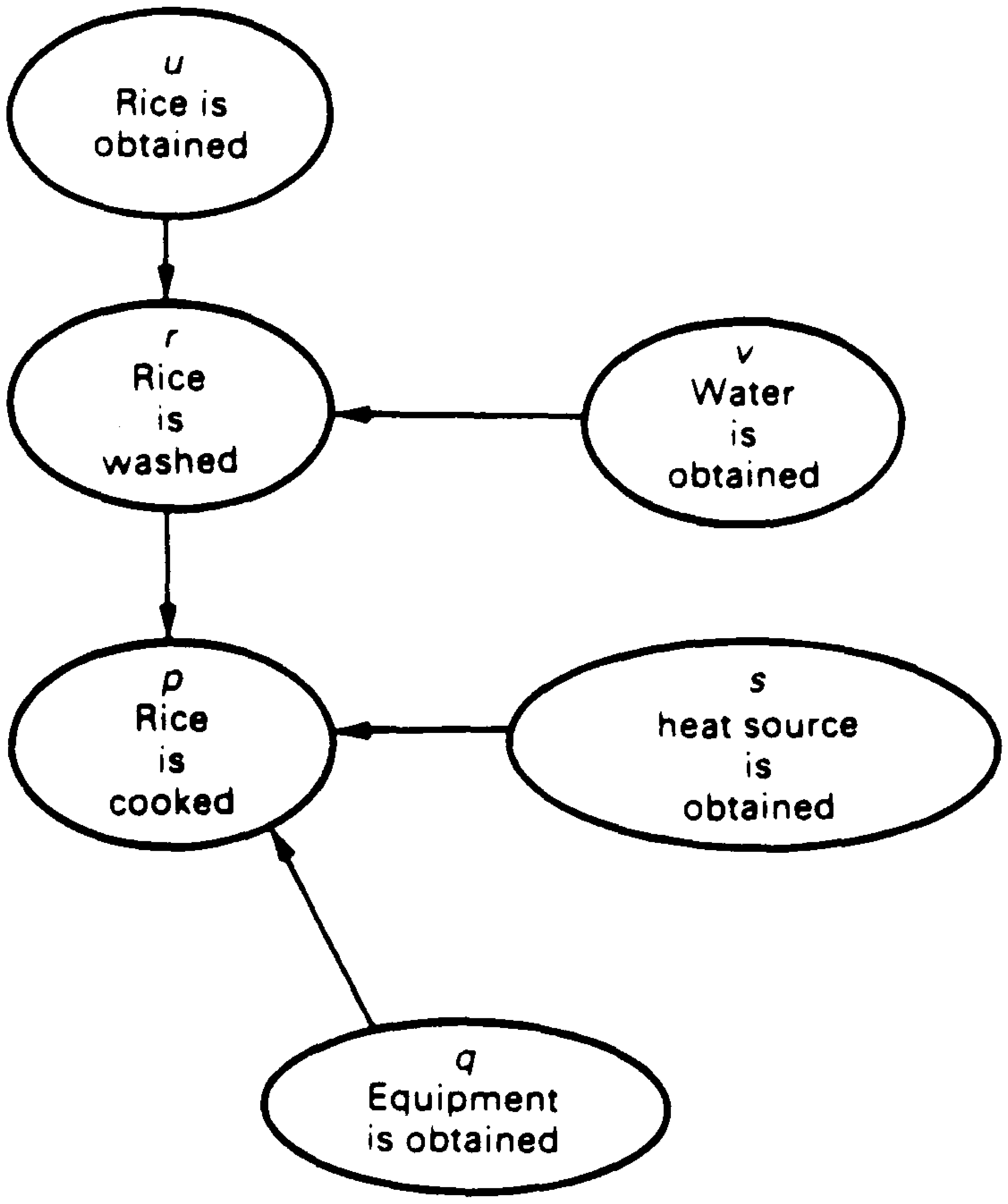

u
Rice is
obtained
r
Rice
is
washed
v
Water
is
obtained
p
Rice
is
cooked
s
heat source
is
obtained
q
Equipment
is obtained

Figure 4

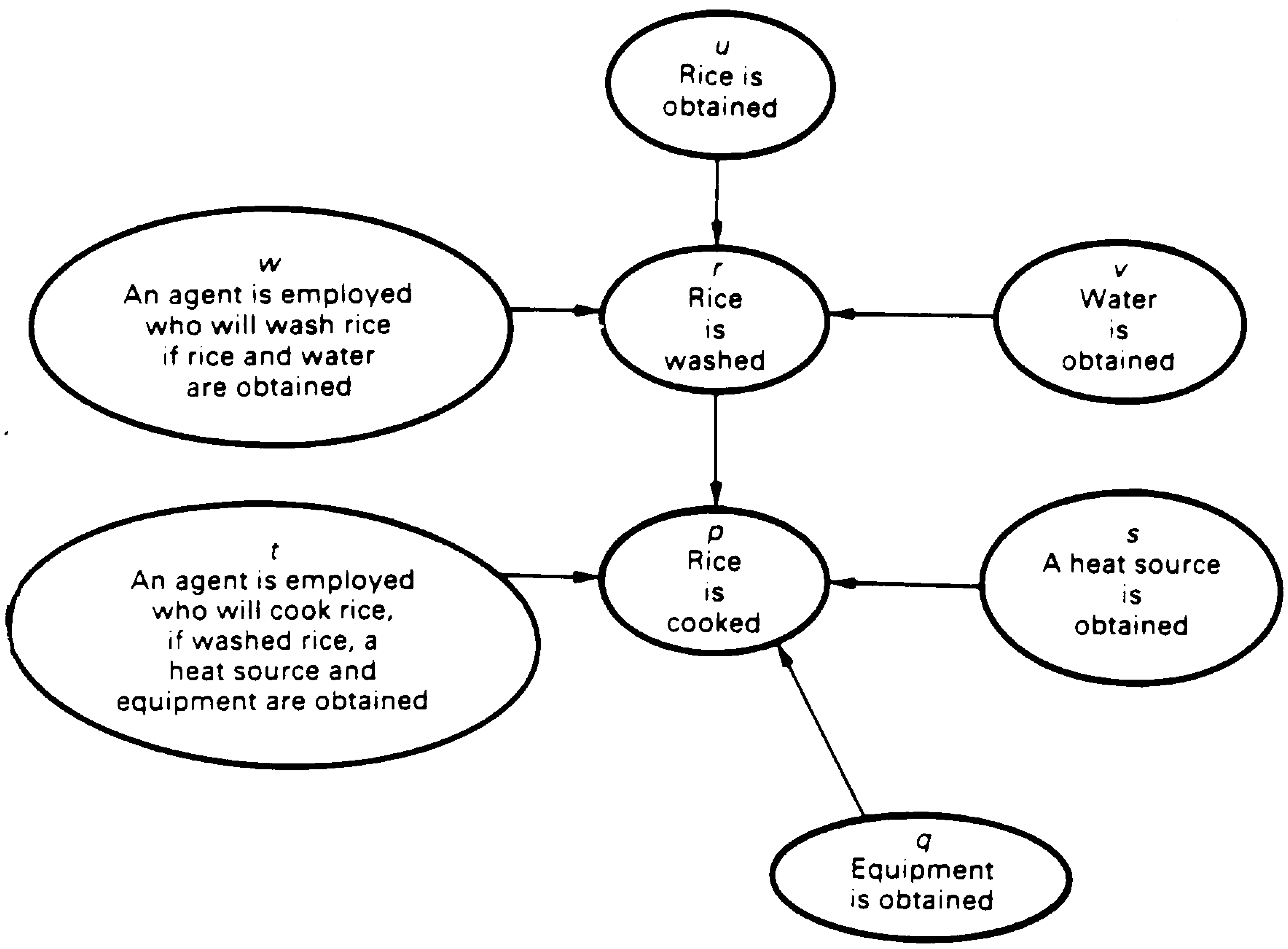

u
Rice is
obtained
w
An agent is employed
who will wash rice
if rice and water
are obtained
r
Rice
is
washed
v
Water
is
obtained
t
An agent is employed
who will cook rice,
if washed rice, a
heat source and
equipment are obtained
p
Rice
is
cooked
s
A heat source
is
obtained
q
Equipment
is obtained

Figure 5

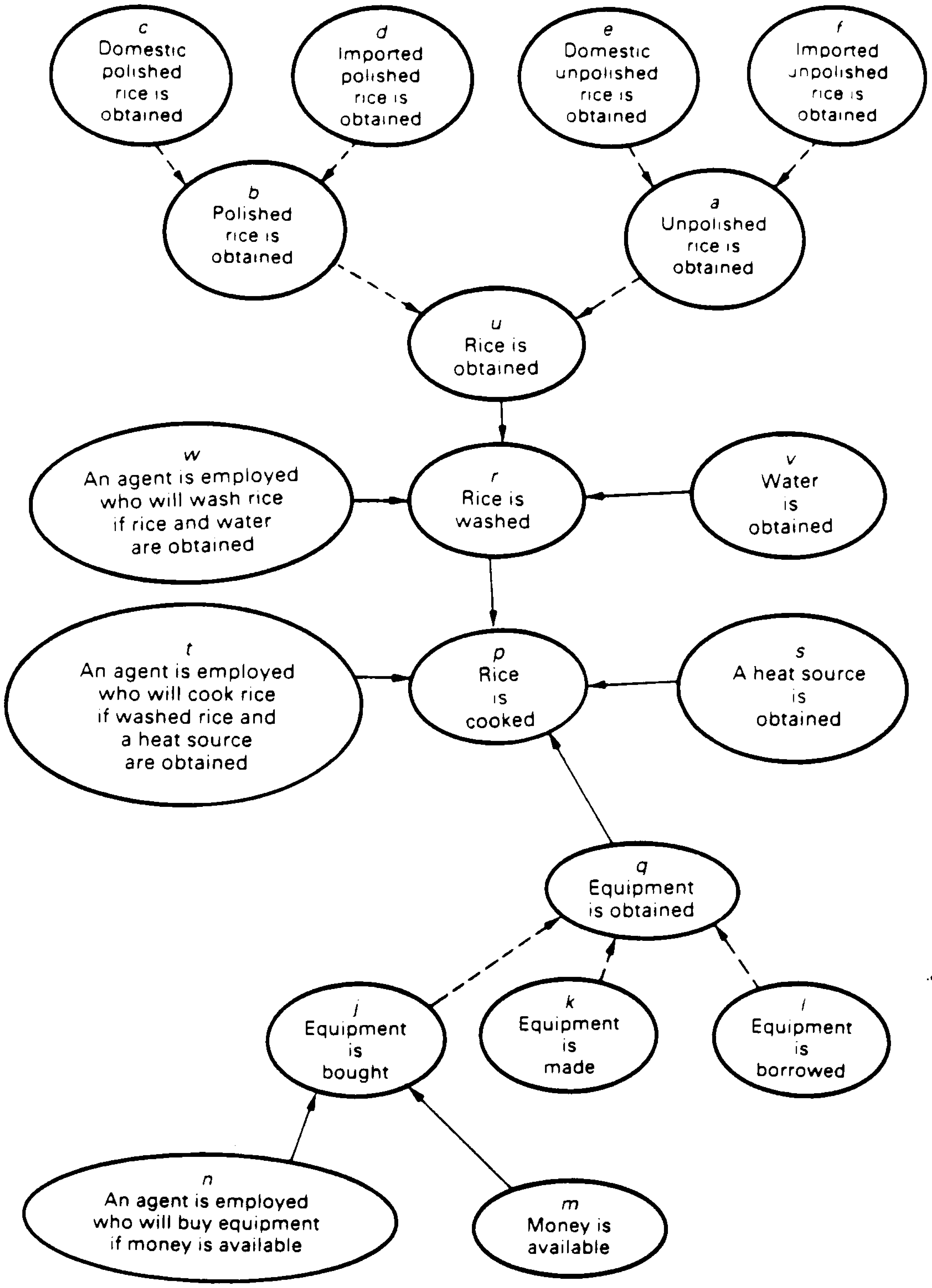
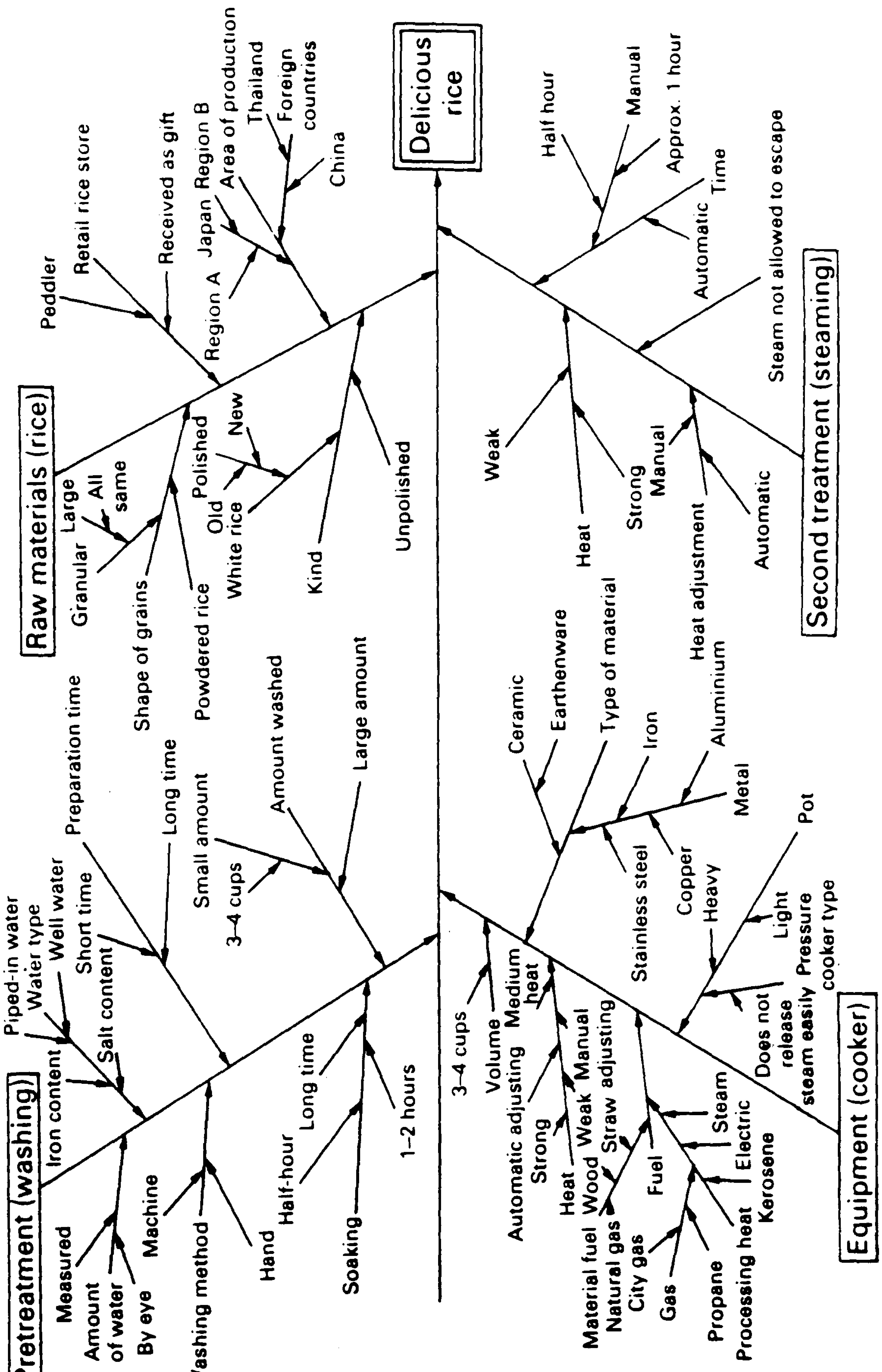

c
Domestic polished rice is obtained
d
Imported polished rice is obtained
e
Domestic unpolished rice is obtained
f
Imported unpolished rice is obtained
b
Polished rice is obtained
a
Unpolished rice is obtained
u
Rice is obtained
w
An agent is employed who will wash rice if rice and water are obtained
r
Rice is washed
v
Water is obtained
t
An agent is employed who will cook rice if washed rice and a heat source are obtained
p
Rice is cooked
s
A heat source is obtained
q
Equipment is obtained
j
Equipment is bought
k
Equipment is made
l
Equipment is borrowed
n
An agent is employed who will buy equipment if money is available
m
Money is available

Figure 6



Delicious rice
Raw materials (rice)
Peddler
Retail rice store
Received as gift
Region A
Japan
Region B
Area of production
Thailand
Foreign countries
China
Granular
Large
All same
Shape of grains
Powdered rice
Polished
New
Old
White rice
Kind
Unpolished
Second treatment (steaming)
Half hour
Manual
Approx. 1 hour
Time
Automatic
Steam not allowed to escape
Weak
Heat
Strong
Manual
Heat adjustment
Automatic
Pretreatment (washing)
Piped-in water
Water type
Well water
Iron content
Salt content
Short time
Preparation time
Long time
Small amount
Amount washed
Large amount
3–4 cups
Measured
Amount of water
By eye
Machine
Washing method
Hand
Half-hour
Long time
Soaking
1–2 hours
Equipment (cooker)
Ceramic
Earthenware
Type of material
Iron
Aluminium
Metal
Stainless steel
Copper
Heavy
Pot
Light
Pressure cooker type
Does not release steam easily
3–4 cups
Volume
Medium heat
Automatic adjusting
Strong
Heat
Weak
Manual
Straw adjusting
Wood
Material fuel
Natural gas
City gas
Gas
Fuel
Steam
Electric
Kerosene
Propane
Processing heat

Figure 7

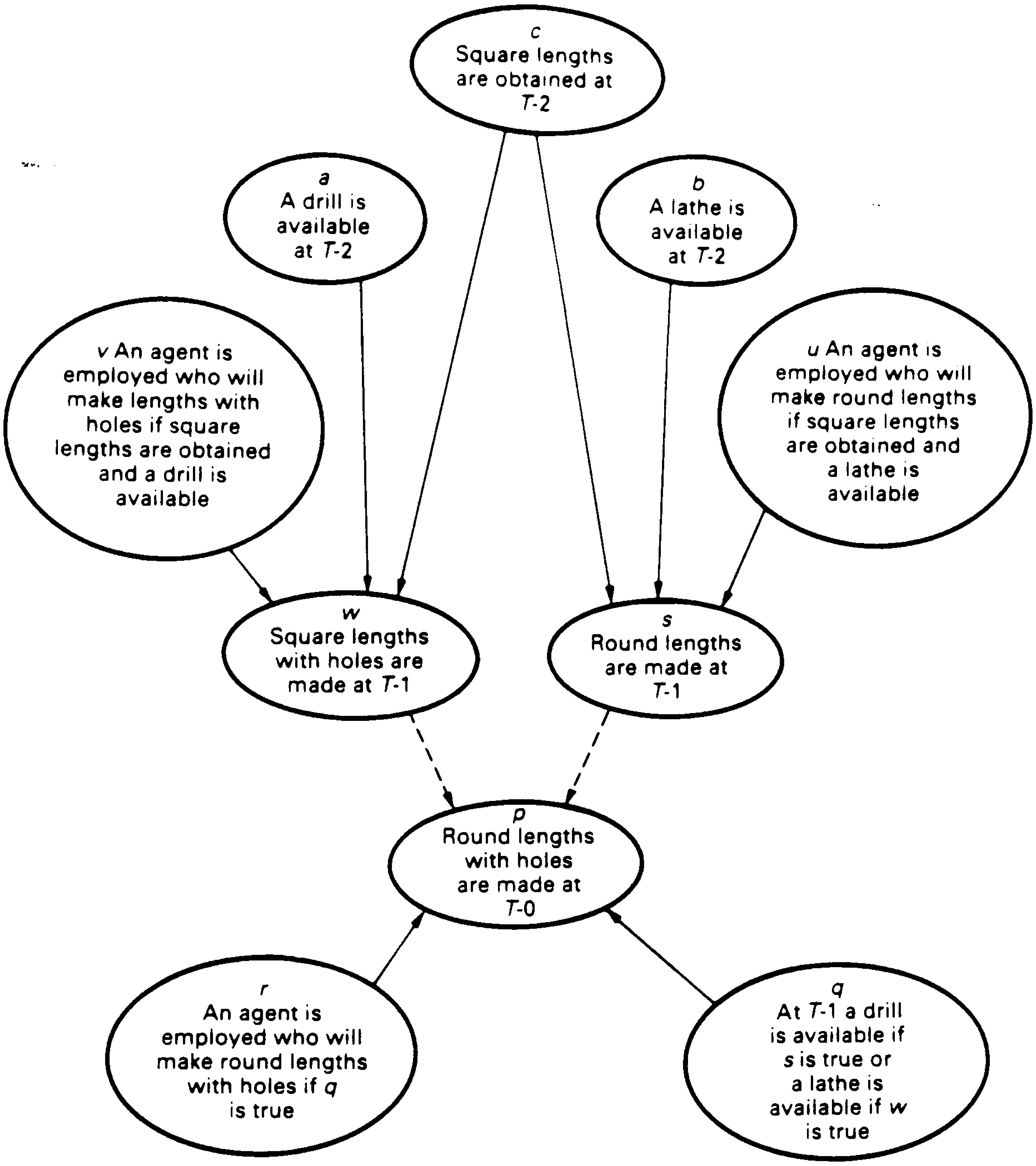
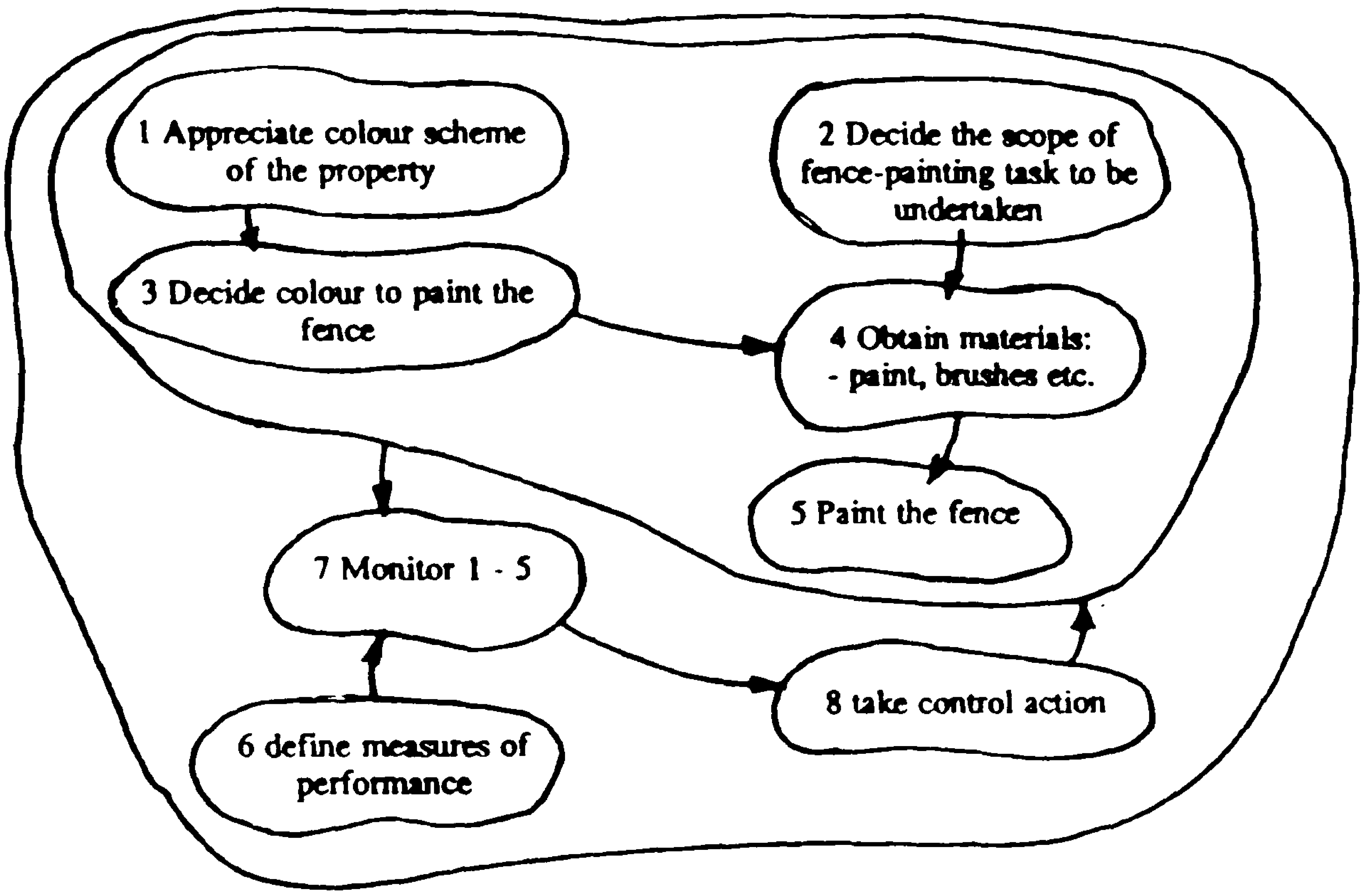

c
Square lengths are obtained at T-2
a
A drill is available at T-2
b
A lathe is available at T-2
v An agent is employed who will make lengths with holes if square lengths are obtained and a drill is available
u An agent is employed who will make round lengths if square lengths are obtained and a lathe is available
w
Square lengths with holes are made at T-1
s
Round lengths are made at T-1
p
Round lengths with holes are made at T-0
r
An agent is employed who will make round lengths with holes if q is true
q
At T-1 a drill is available if s is true or a lathe is available if w is true

Figure 8



1 Appreciate colour scheme of the property
2 Decide the scope of fence-painting task to be undertaken
3 Decide colour to paint the fence
4 Obtain materials: - paint, brushes etc.
5 Paint the fence
7 Monitor 1 - 5
6 define measures of performance
8 take control action

Figure 9

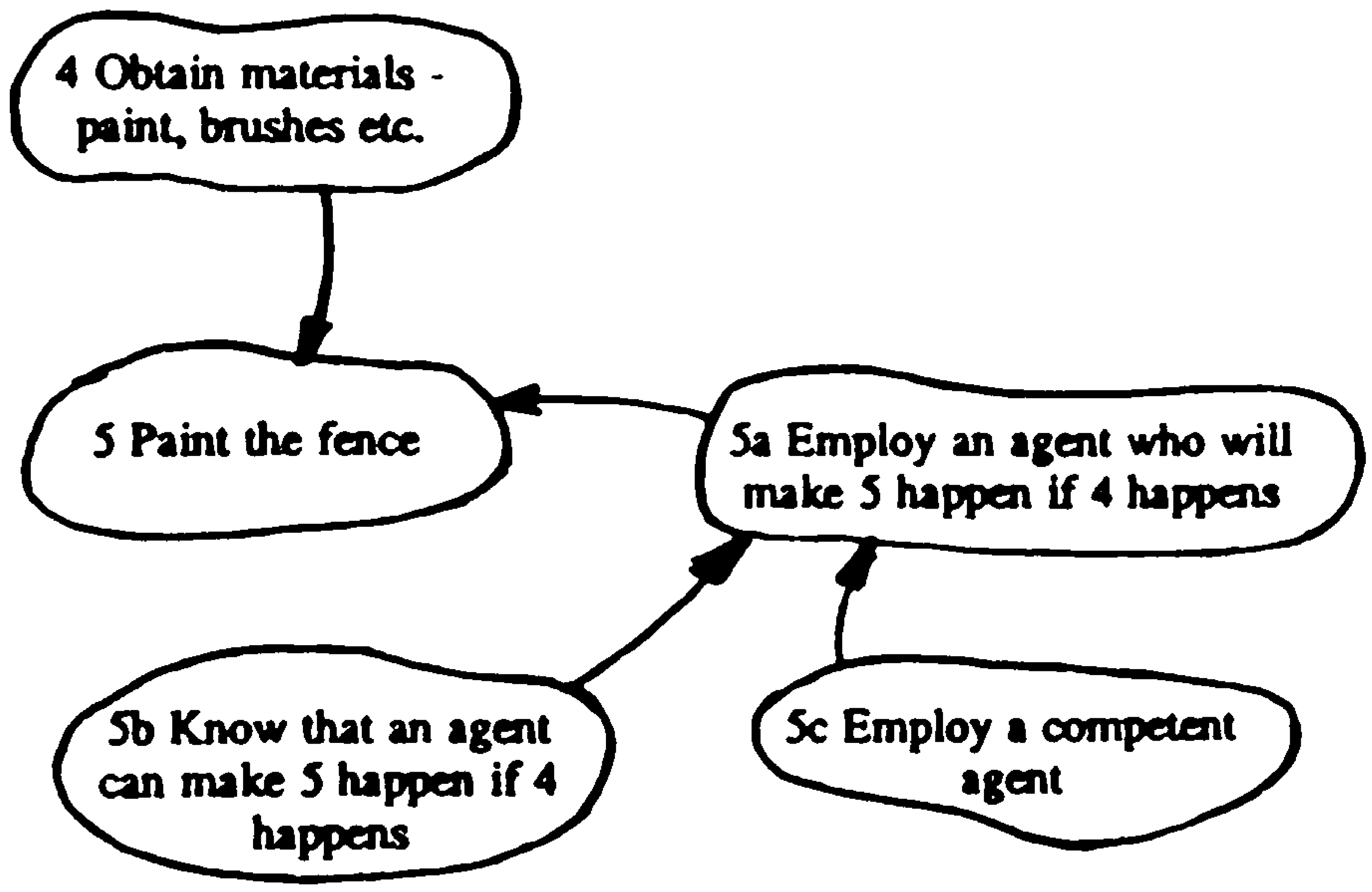

4 Obtain materials - paint, brushes etc.
5 Paint the fence
5a Employ an agent who will make 5 happen if 4 happens
5b Know that an agent can make 5 happen if 4 happens
5c Employ a competent agent

Figure 10

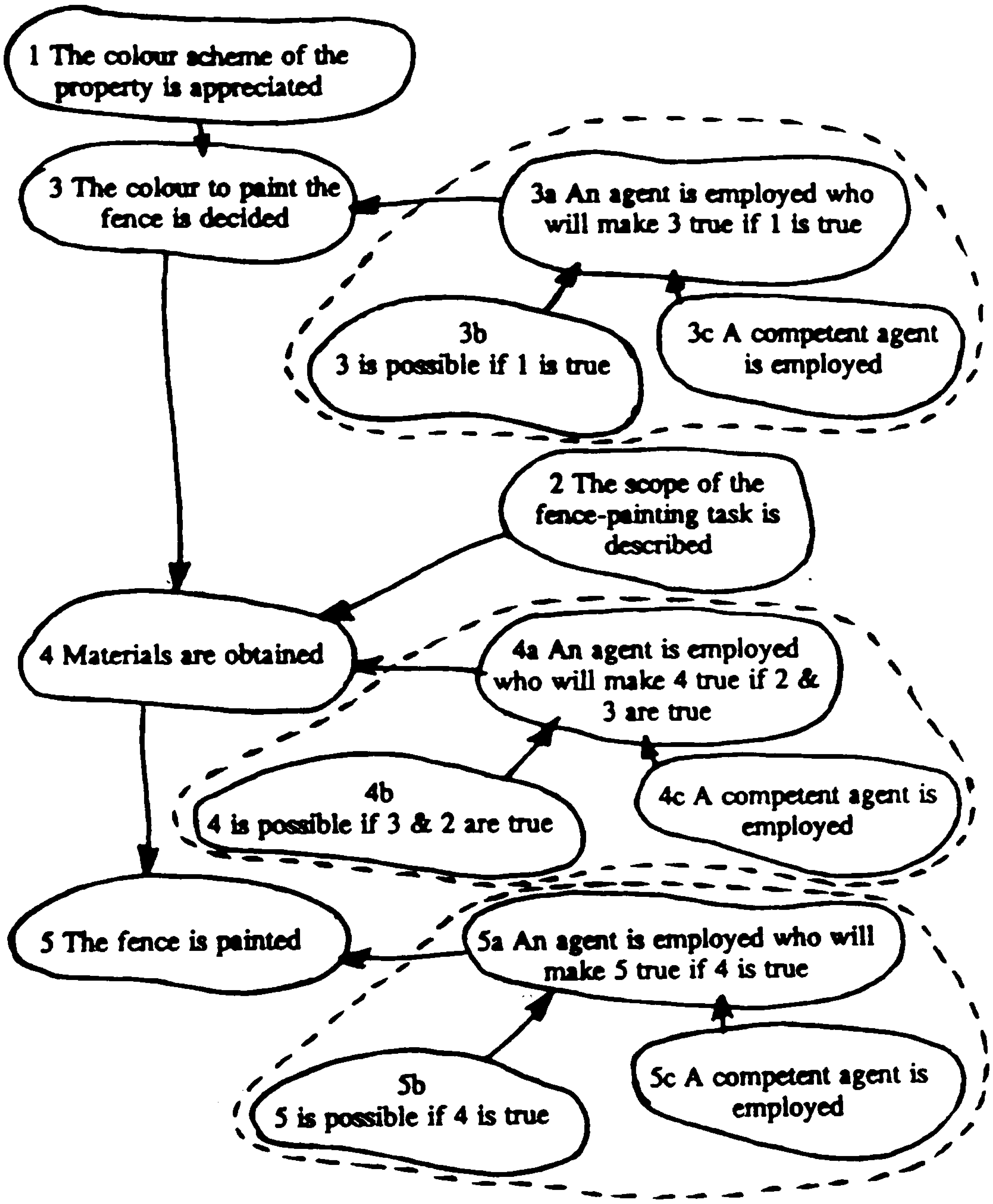
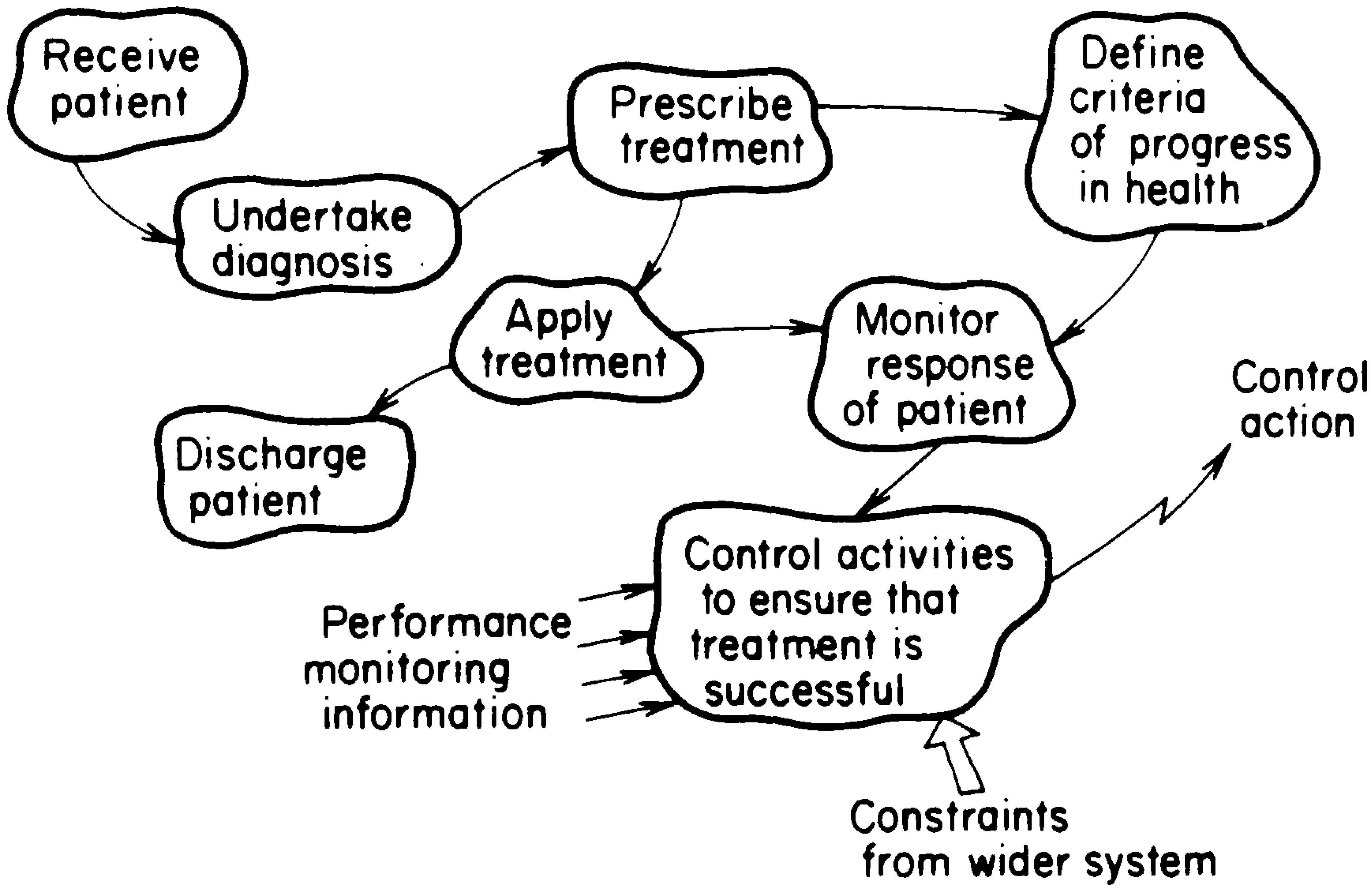

1 The colour scheme of the property is appreciated
3 The colour to paint the fence is decided
3a An agent is employed who will make 3 true if 1 is true
3b
3 is possible if 1 is true
3c A competent agent is employed
2 The scope of the fence-painting task is described
4 Materials are obtained
4a An agent is employed who will make 4 true if 2 & 3 are true
4b
4 is possible if 3 & 2 are true
4c A competent agent is employed
5 The fence is painted
5a An agent is employed who will make 5 true if 4 is true
5b
5 is possible if 4 is true
5c A competent agent is employed

Figure 11



Receive patient
Undertake diagnosis
Prescribe treatment
Define criteria of progress in health
Apply treatment
Monitor response of patient
Discharge patient
Control action
Performance monitoring information
Control activities to ensure that treatment is successful
Constraints from wider system

Figure 12

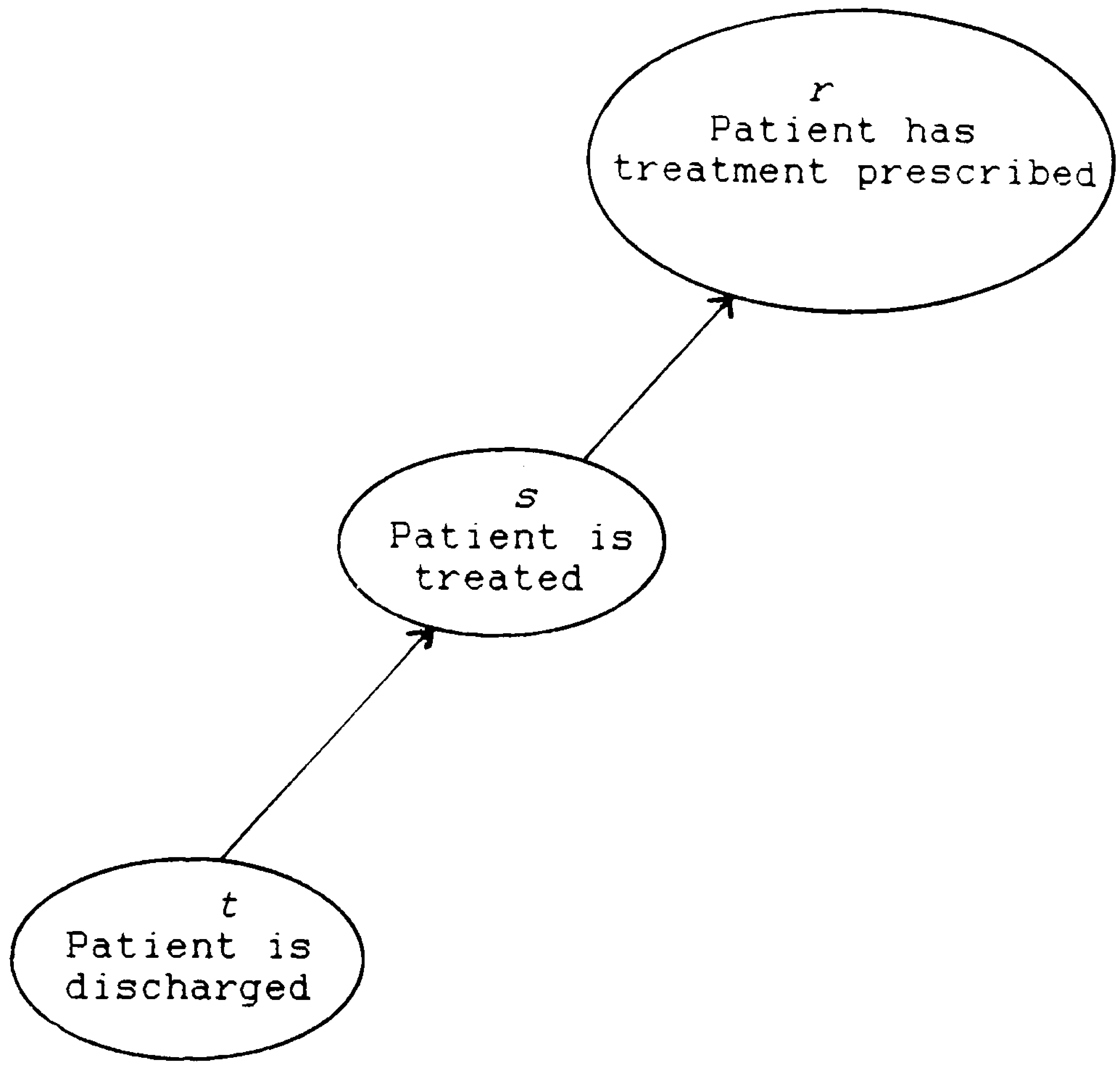

r
Patient has
treatment prescribed
s
Patient is
treated
t
Patient is
discharged

Figure 13

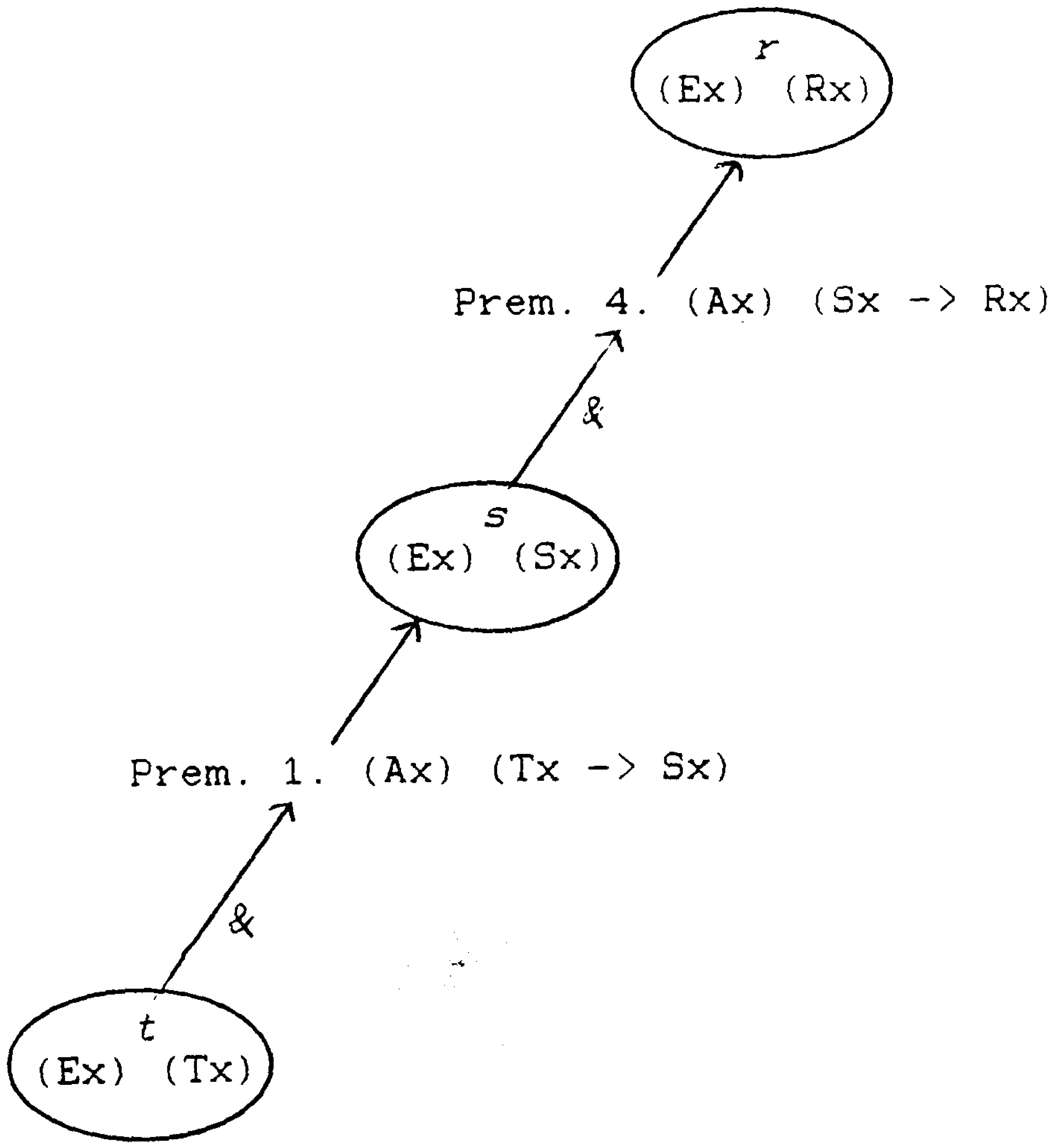


Figure 14

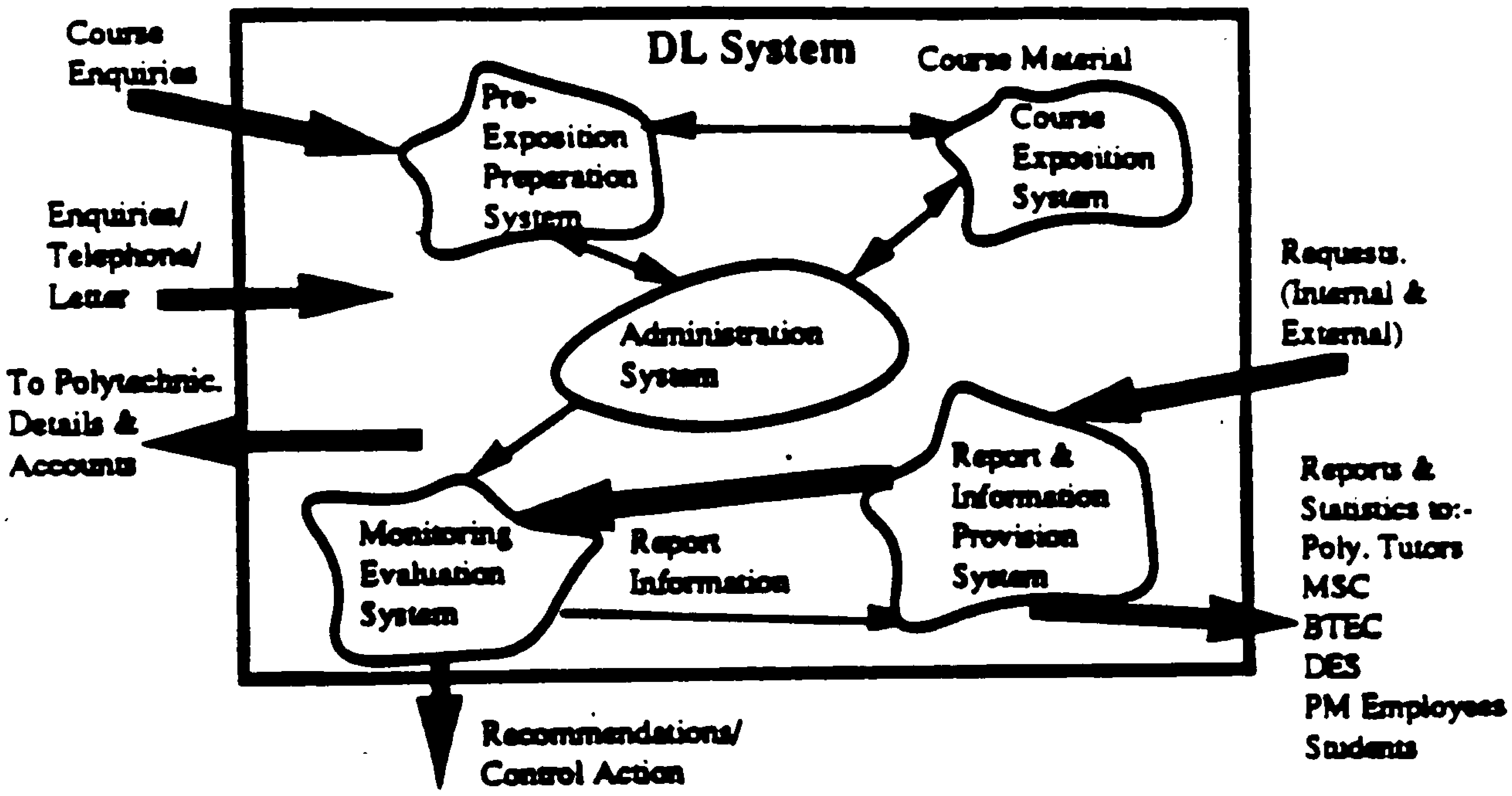

DL System
Course Enquiries
Pre-Exposition Preperation System
Course Material
Course Exposition System
Enquiries/ Telephone/ Letter
Administration System
Requests. (Internal & External)
To Polytechnic. Details & Accounts
Report & Information Provision System
Monitoring Evaluation System
Report Information
Reports & Statistics to:- Poly. Tutors MSC BTEC DES PM Employees Students
Recommendations/ Control Action

Figure 15

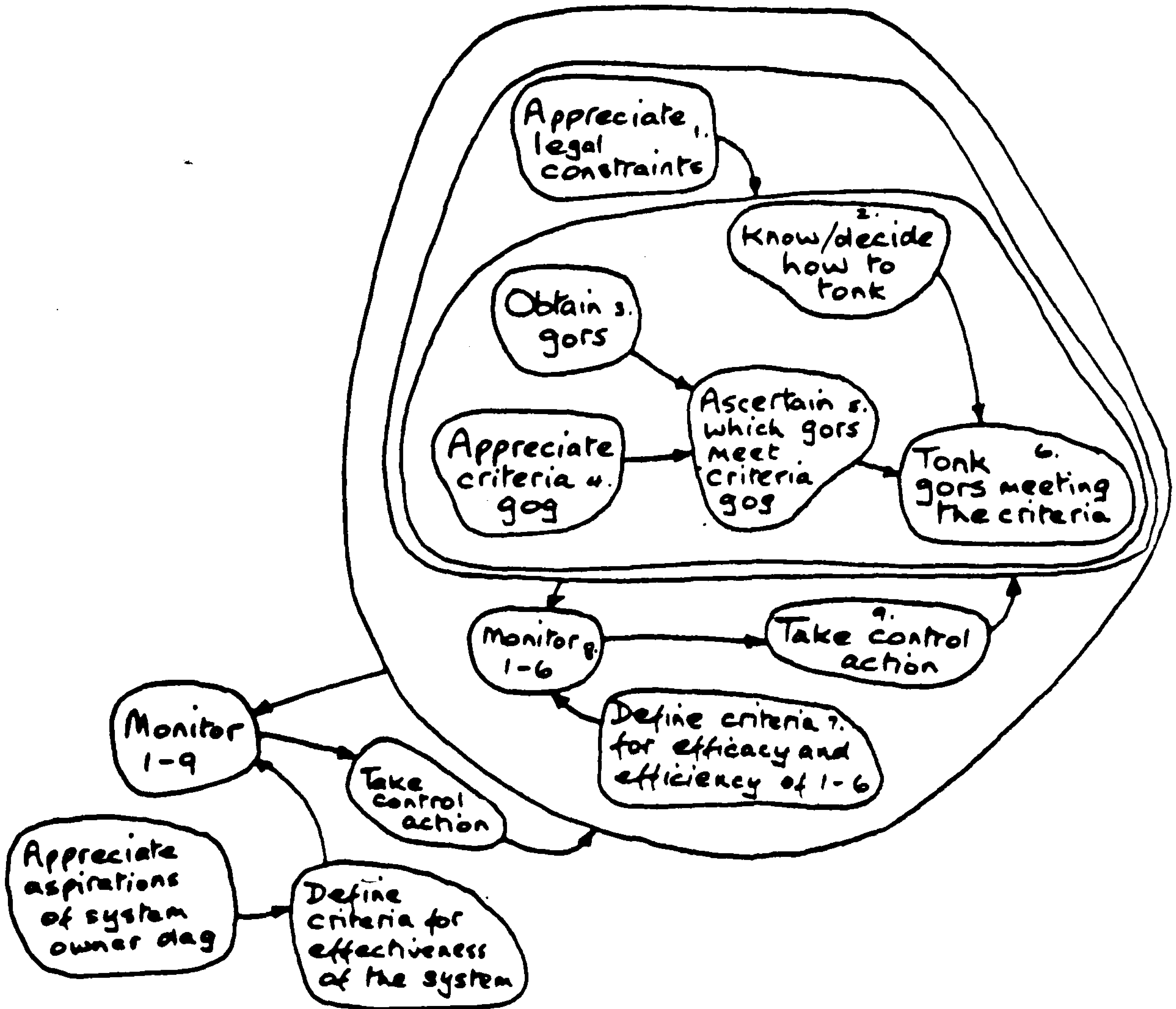
Model Building :

A defensible logical structure for a model may be derived from a root definition even though knowledge of any real-world version of the purposeful activity is lacking!

Root Definition

A dag-owned gor tonking system which, within legal constraints, tonks those gors which meet criteria gog.

C gors
A not stated (skilled tonkers implied)
T gors ⟶ tonked gors
W gor tonking is a good thing to do
O dag
E legal constraints



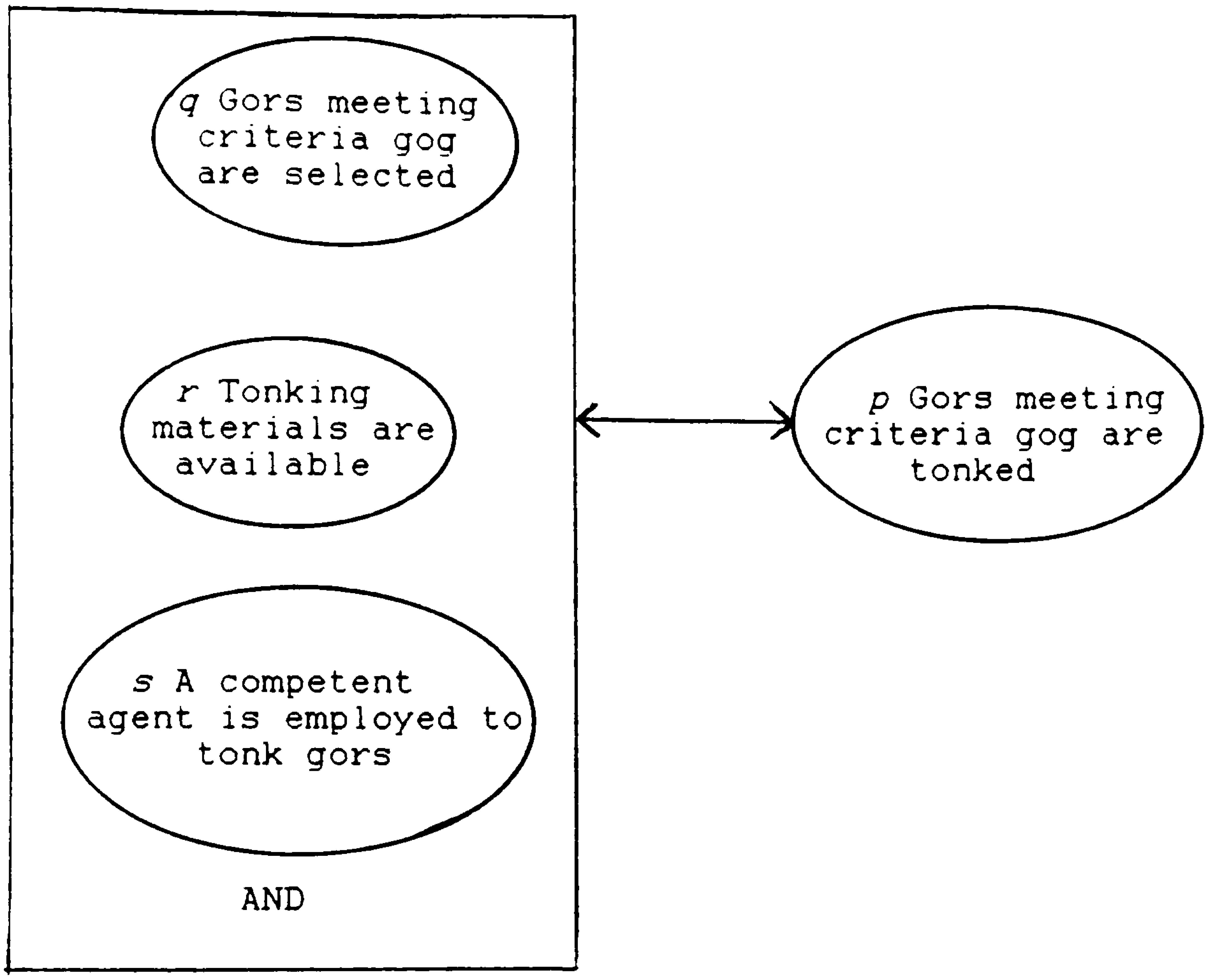

q Gors meeting criteria gog are selected
r Tonking materials are available
s A competent agent is employed to tonk gors
AND
p Gors meeting criteria gog are tonked

Figure 17

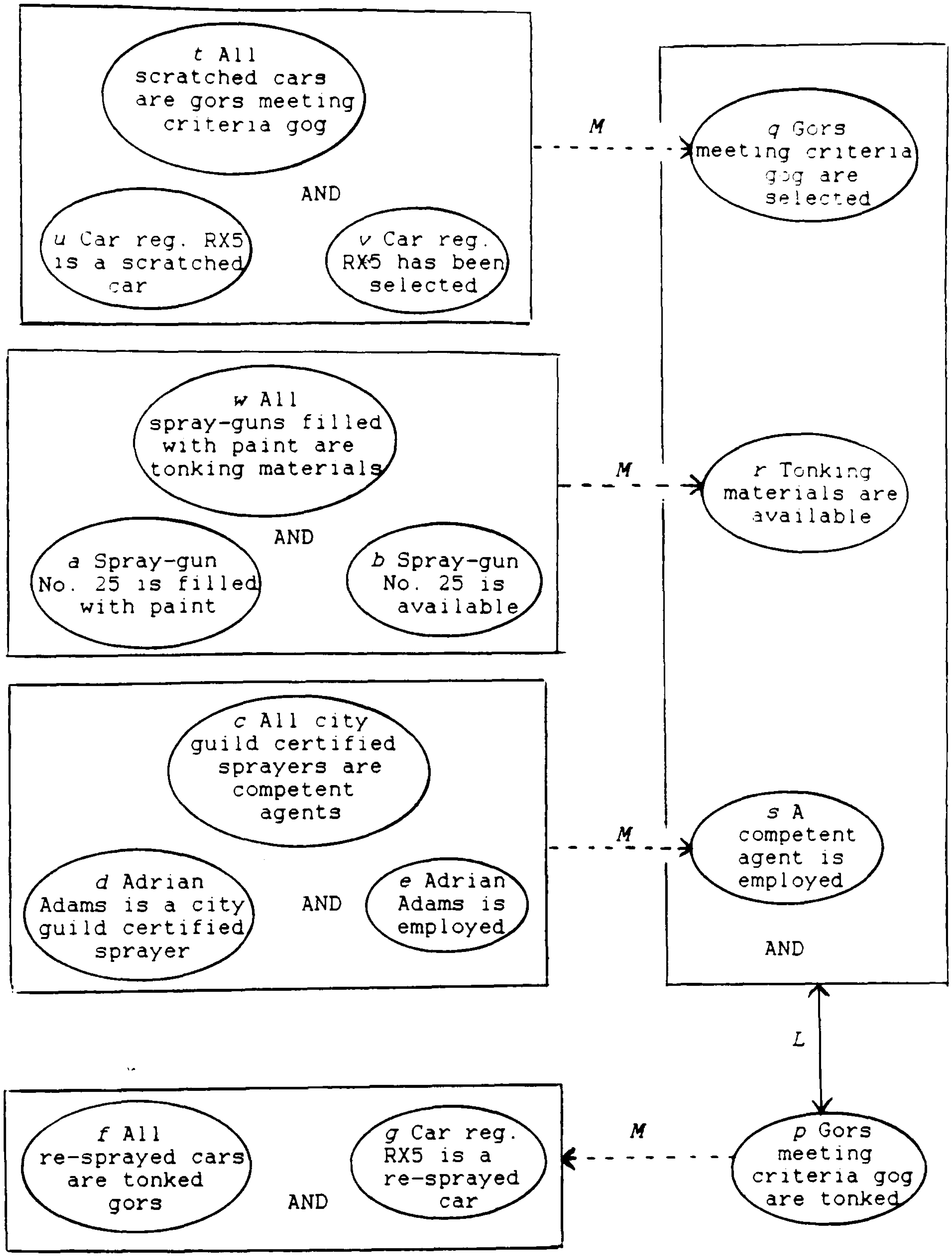

t All scratched cars are gors meeting criteria gog
AND
u Car reg. RX5 is a scratched car
v Car reg. RX5 has been selected
M
q Gors meeting criteria gog are selected
w All spray-guns filled with paint are tonking materials
AND
a Spray-gun No. 25 is filled with paint
b Spray-gun No. 25 is available
M
r Tonking materials are available
c All city guild certified sprayers are competent agents
d Adrian Adams is a city guild certified sprayer
AND
e Adrian Adams is employed
M
s A competent agent is employed
AND
L
f All re-sprayed cars are tonked gors
AND
g Car reg. RX5 is a re-sprayed car
M
p Gors meeting criteria gog are tonked

Figure 18

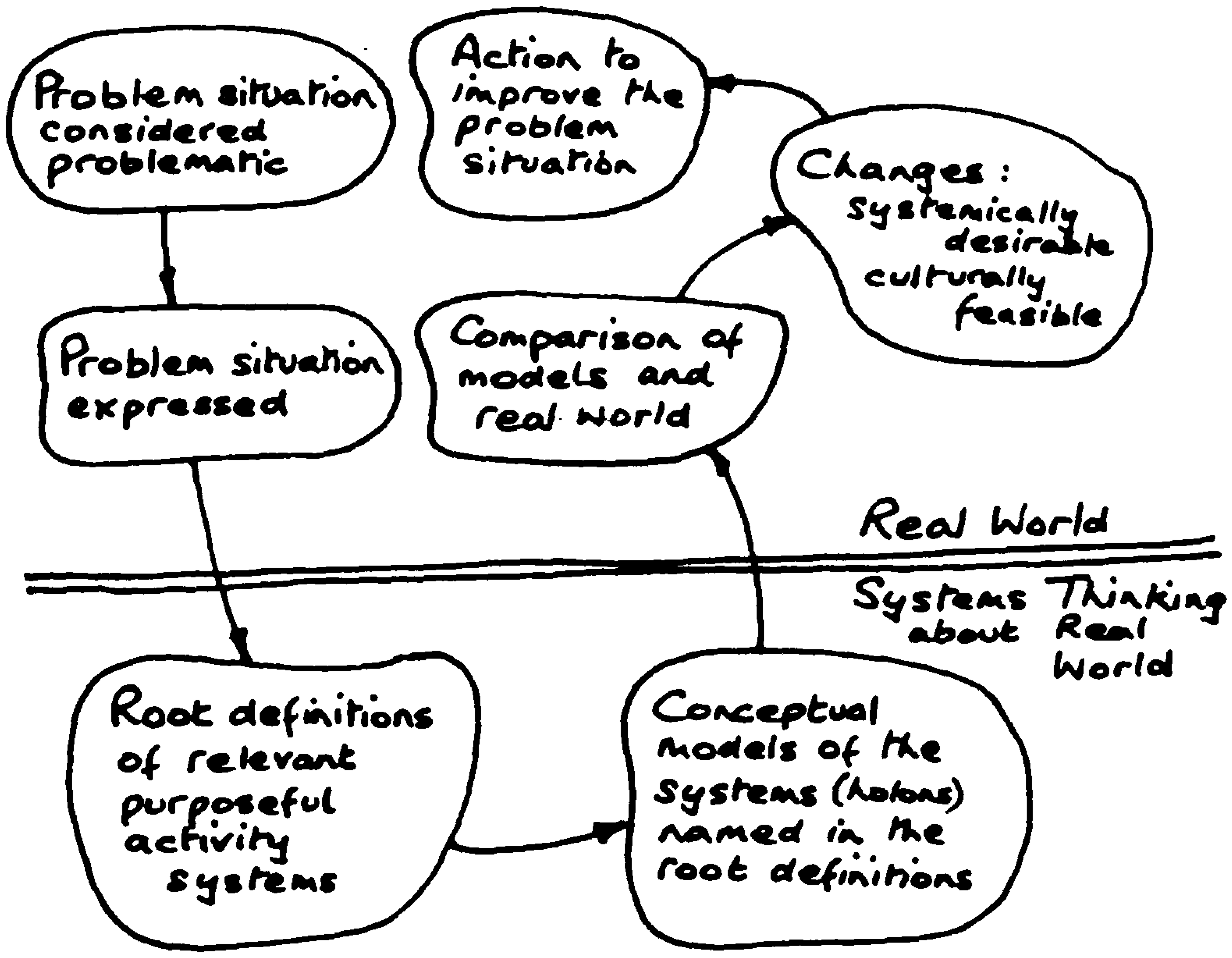

Problem situation considered problematic
Problem situation expressed
Root definitions of relevant purposeful activity systems
Conceptual models of the systems (holons) named in the root definitions
Comparison of models and real world
Changes: systemically desirable culturally feasible
Action to improve the problem situation
Real World
Systems Thinking about Real World

Figure 19

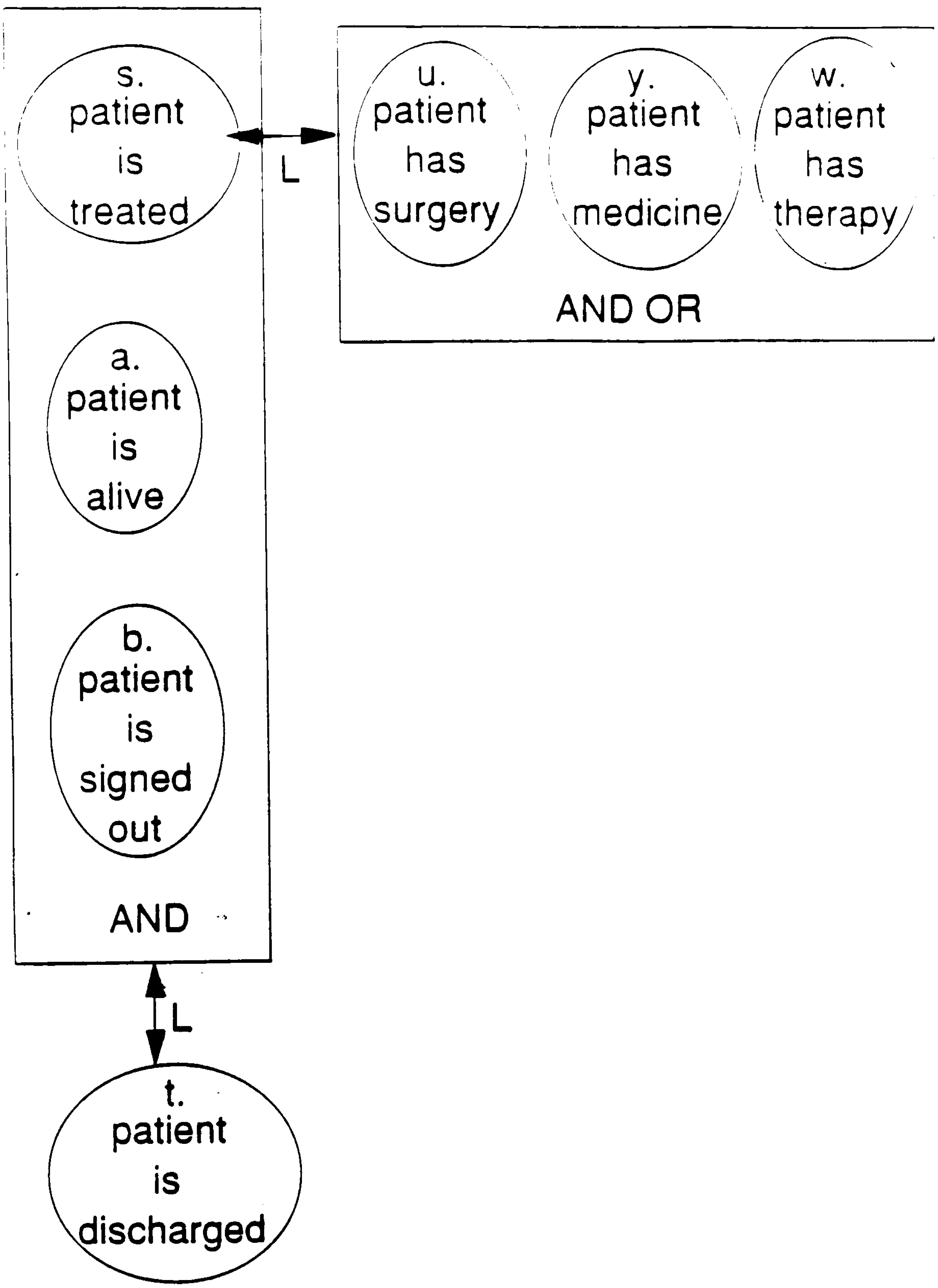

s.
patient
is
treated
L
u.
patient
has
surgery
y.
patient
has
medicine
w.
patient
has
therapy
AND OR
a.
patient
is
alive
b.
patient
is
signed
out
AND
L
t.
patient
is
discharged

Figure 20

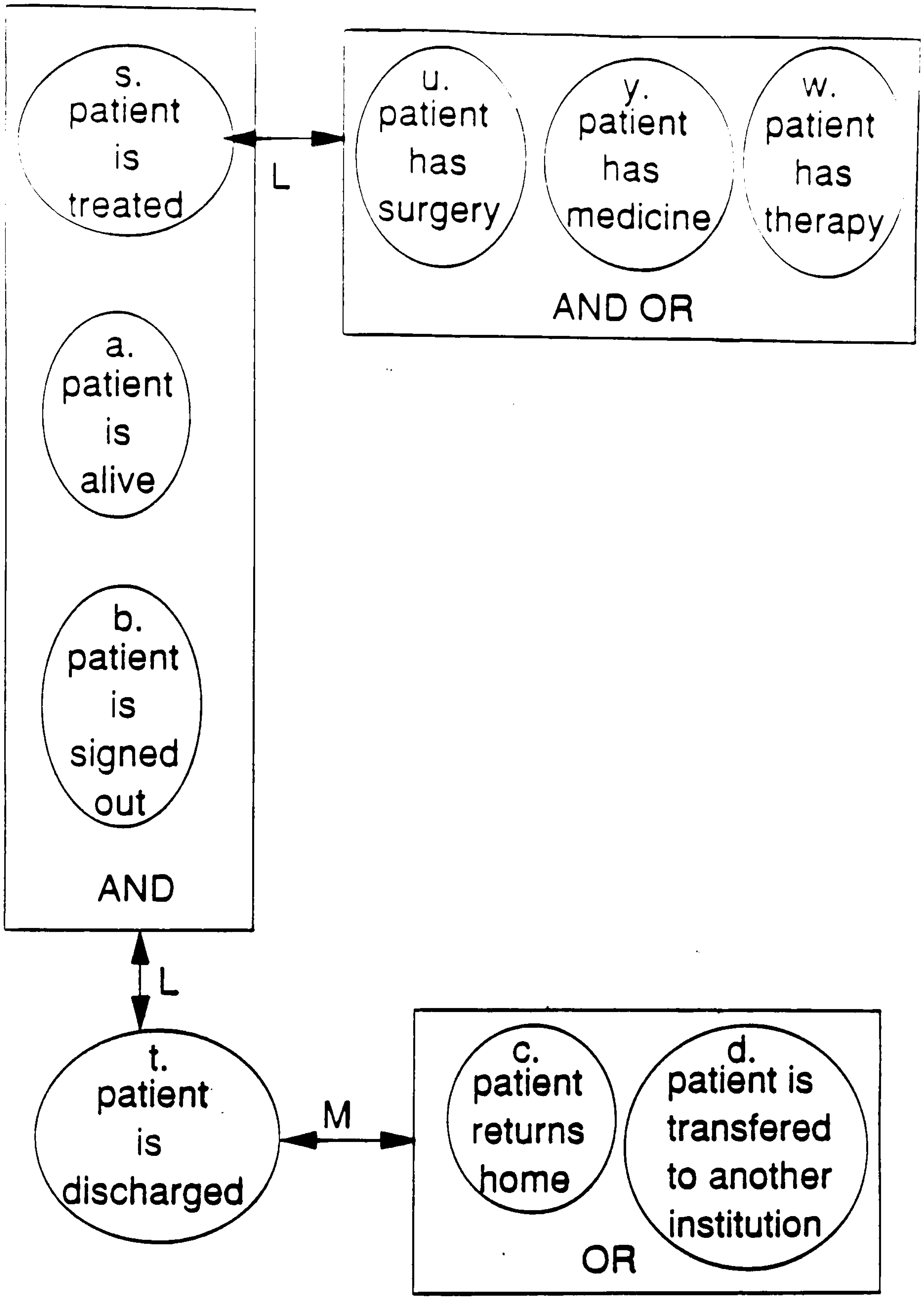
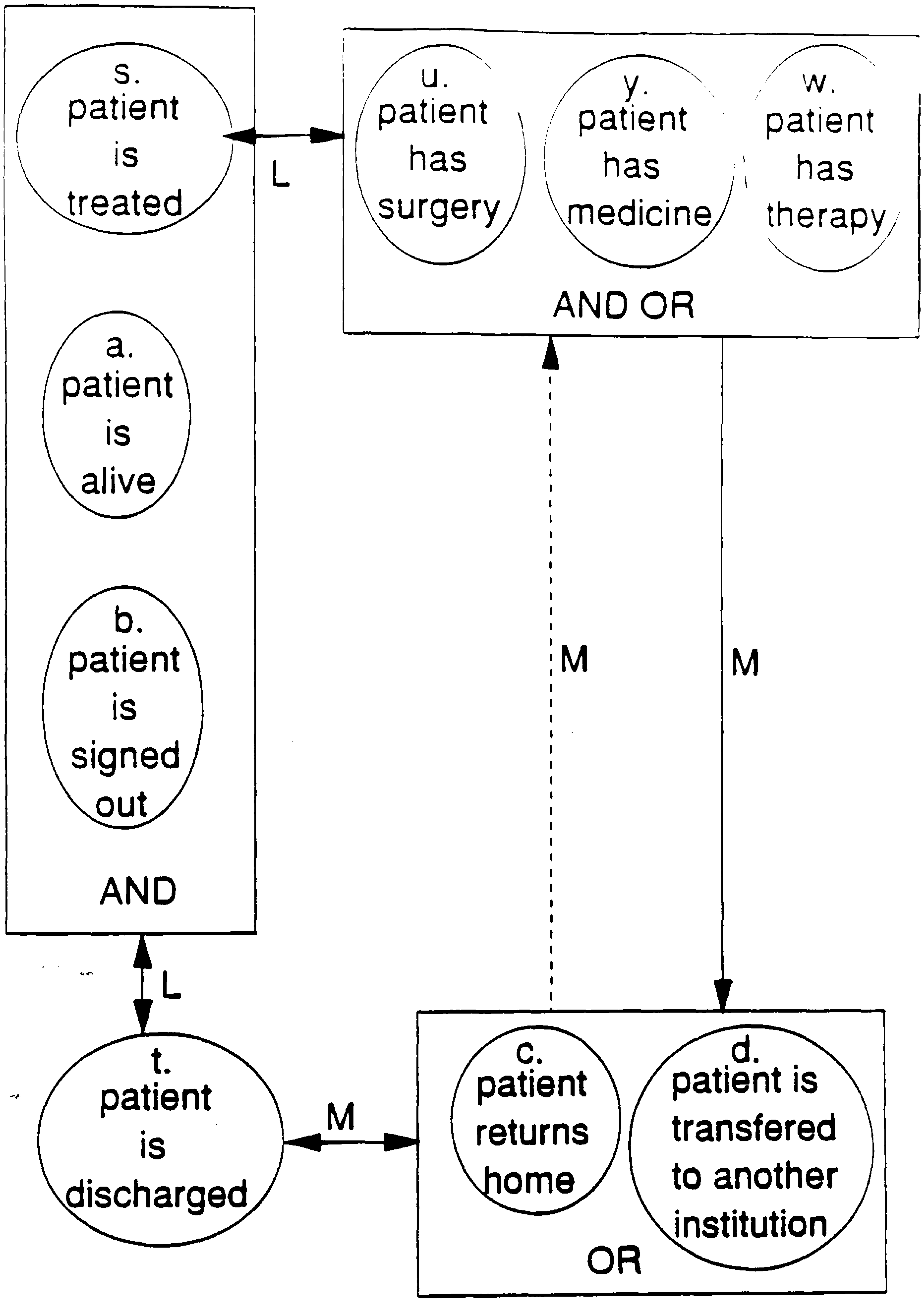
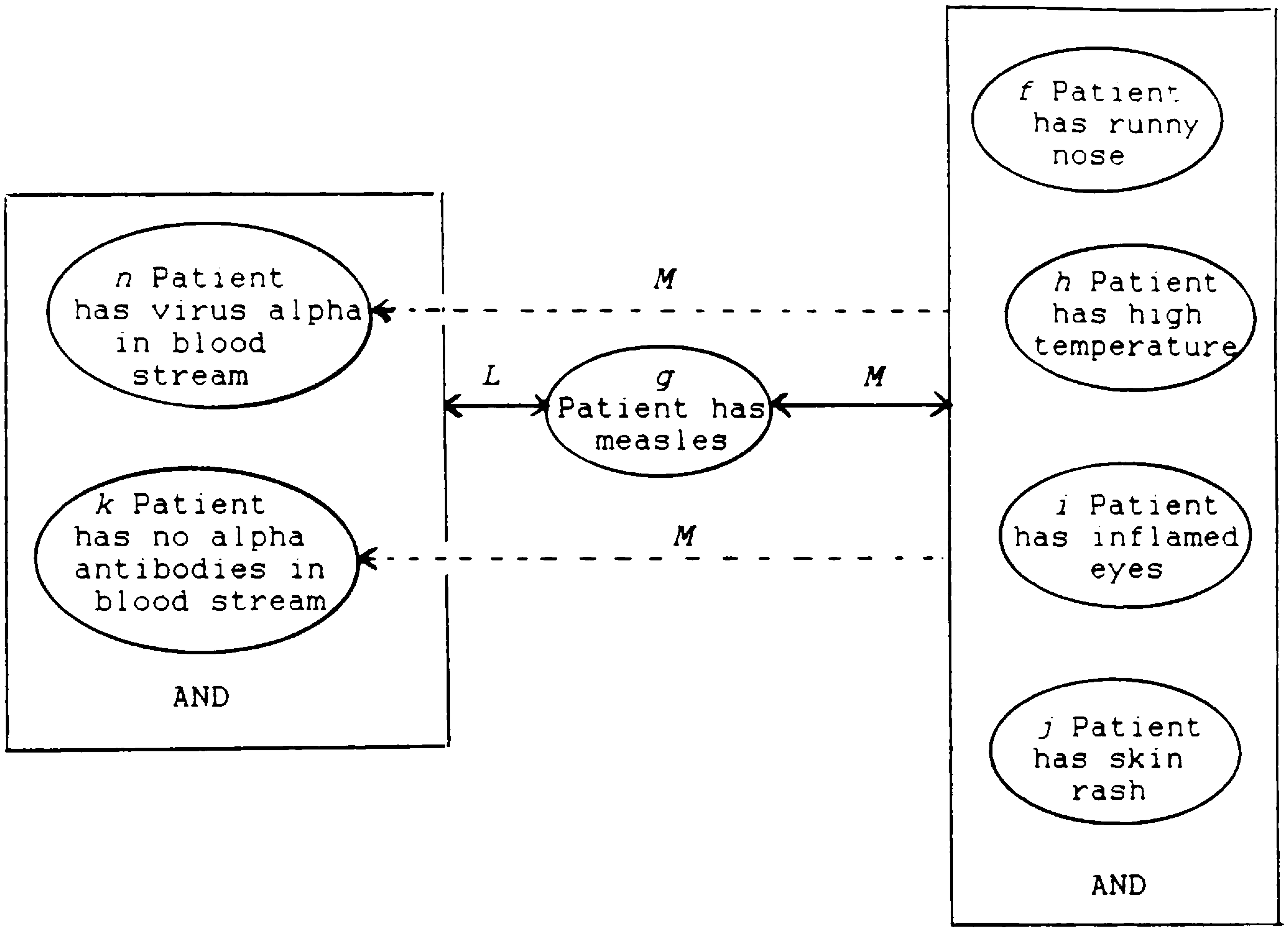

s. patient is treated
u. patient has surgery
y. patient has medicine
w. patient has therapy
AND OR
L
a. patient is alive
b. patient is signed out
AND
L
t. patient is discharged
M
c. patient returns home
d. patient is transfered to another institution
OR

Figure 21



s. patient is treated
a. patient is alive
b. patient is signed out
AND
L
u. patient has surgery
y. patient has medicine
w. patient has therapy
AND OR
M
M
L
t. patient is discharged
M
c. patient returns home
d. patient is transfered to another institution
OR

Figure 22



n Patient has virus alpha in blood stream
k Patient has no alpha antibodies in blood stream
AND
L
g Patient has measles
M
M
M
f Patient has runny nose
h Patient has high temperature
i Patient has inflamed eyes
j Patient has skin rash
AND

Figure 23

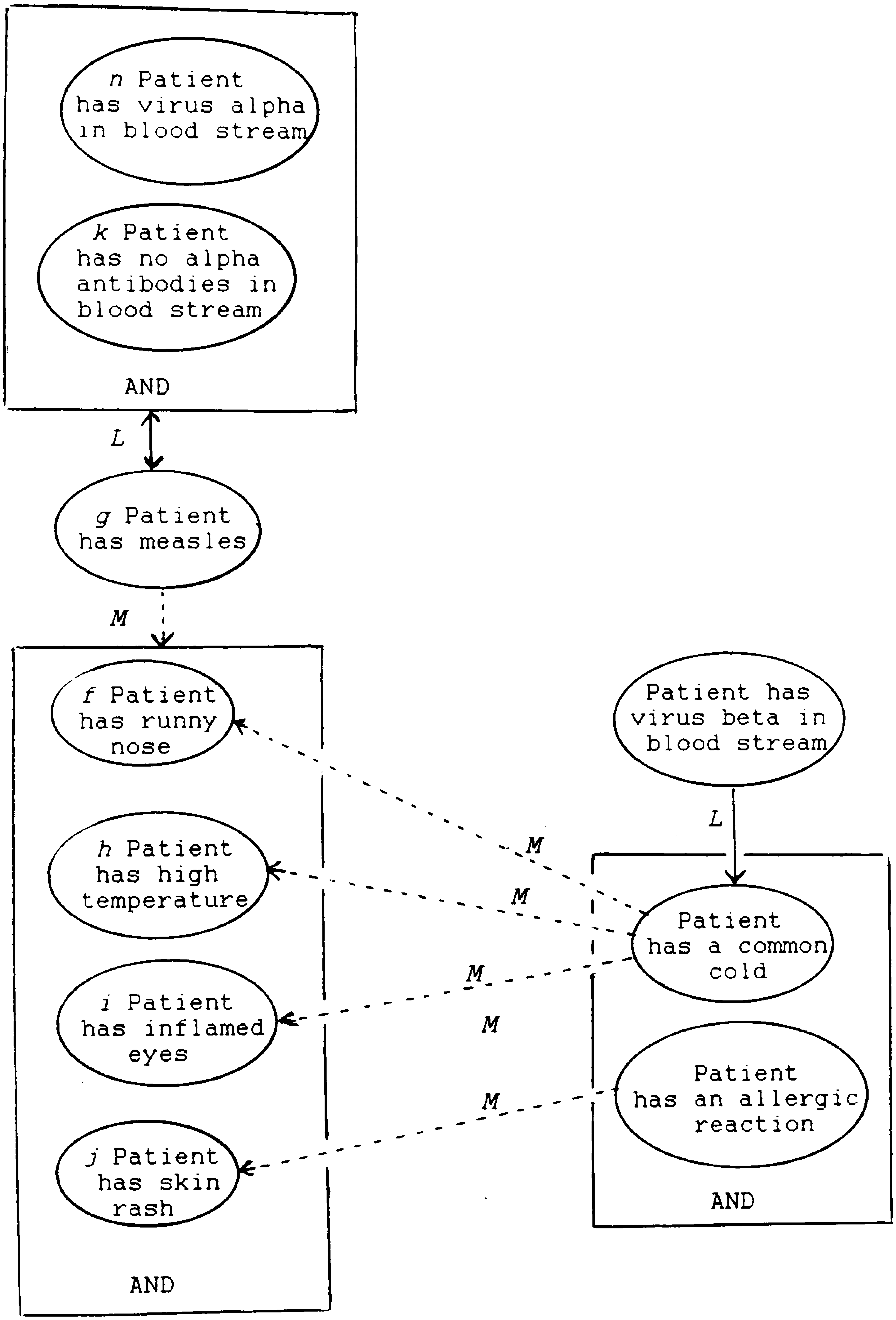

n Patient has virus alpha in blood stream
k Patient has no alpha antibodies in blood stream
AND
L
g Patient has measles
M
f Patient has runny nose
h Patient has high temperature
i Patient has inflamed eyes
j Patient has skin rash
AND
Patient has virus beta in blood stream
L
M
M
M
M
M
Patient has a common cold
Patient has an allergic reaction
AND

Figure 24

Conceptual Model - Level One

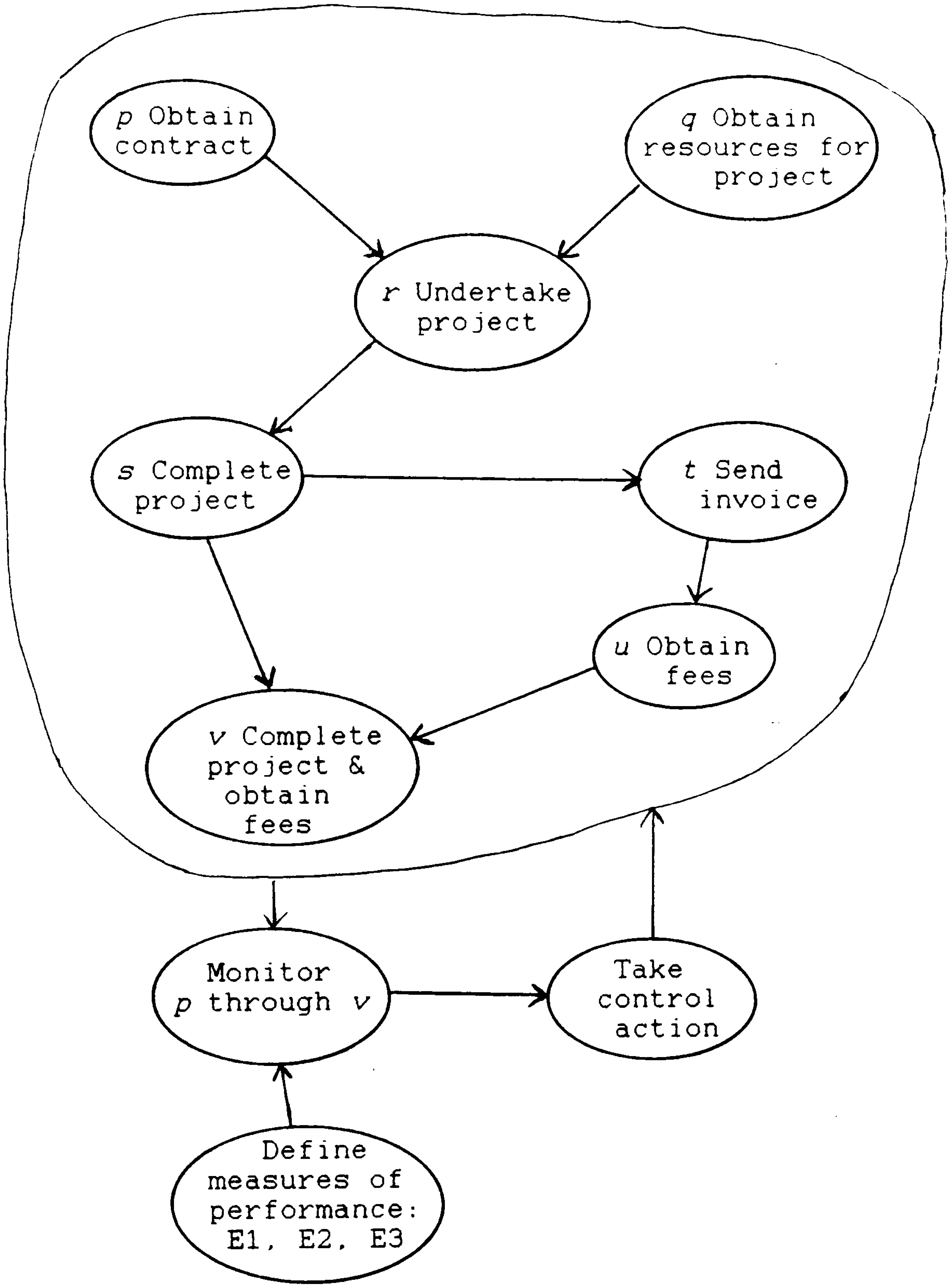


Figure 25

Conceptual Model - Level Two, "Undertake Project"

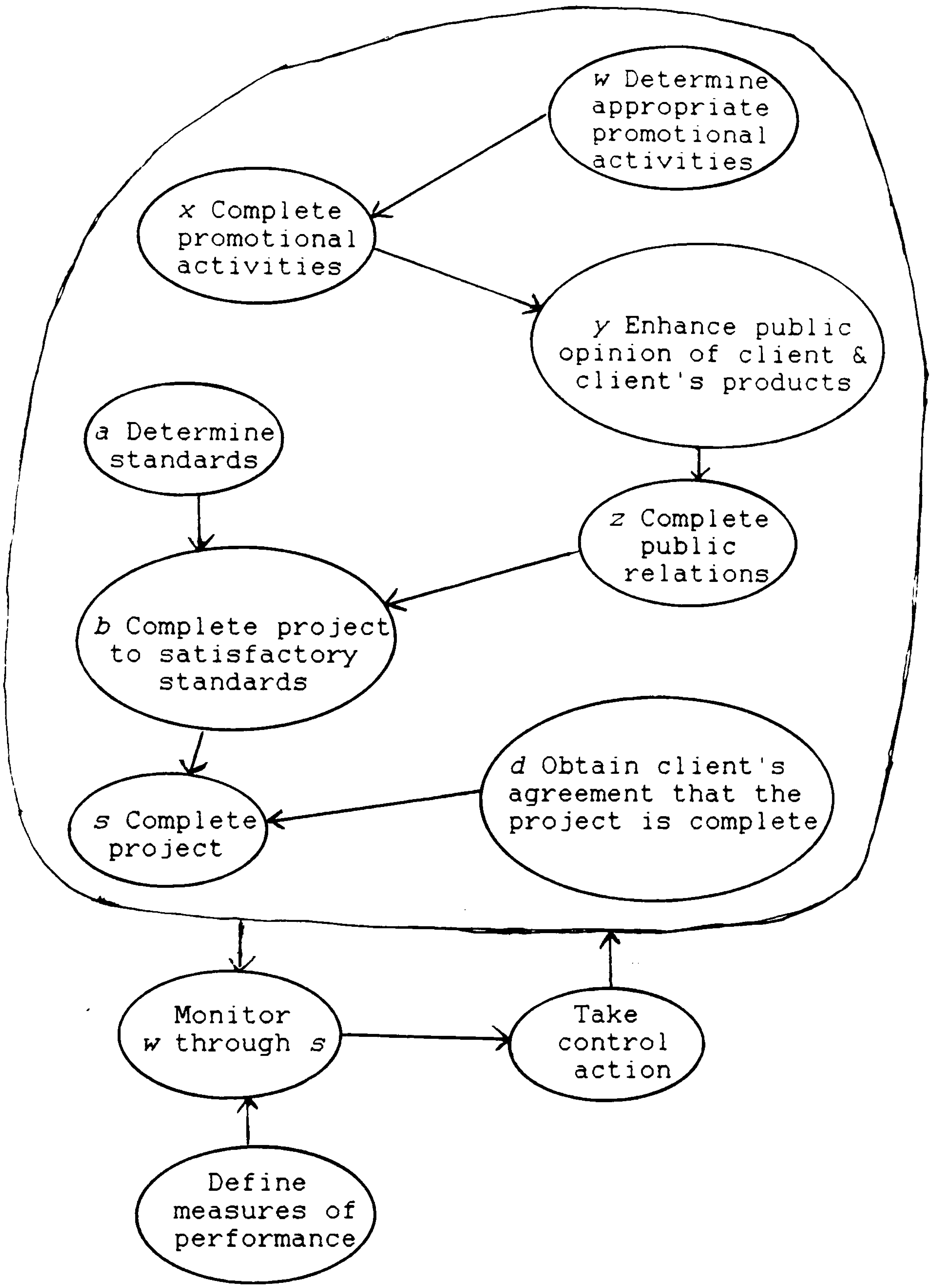
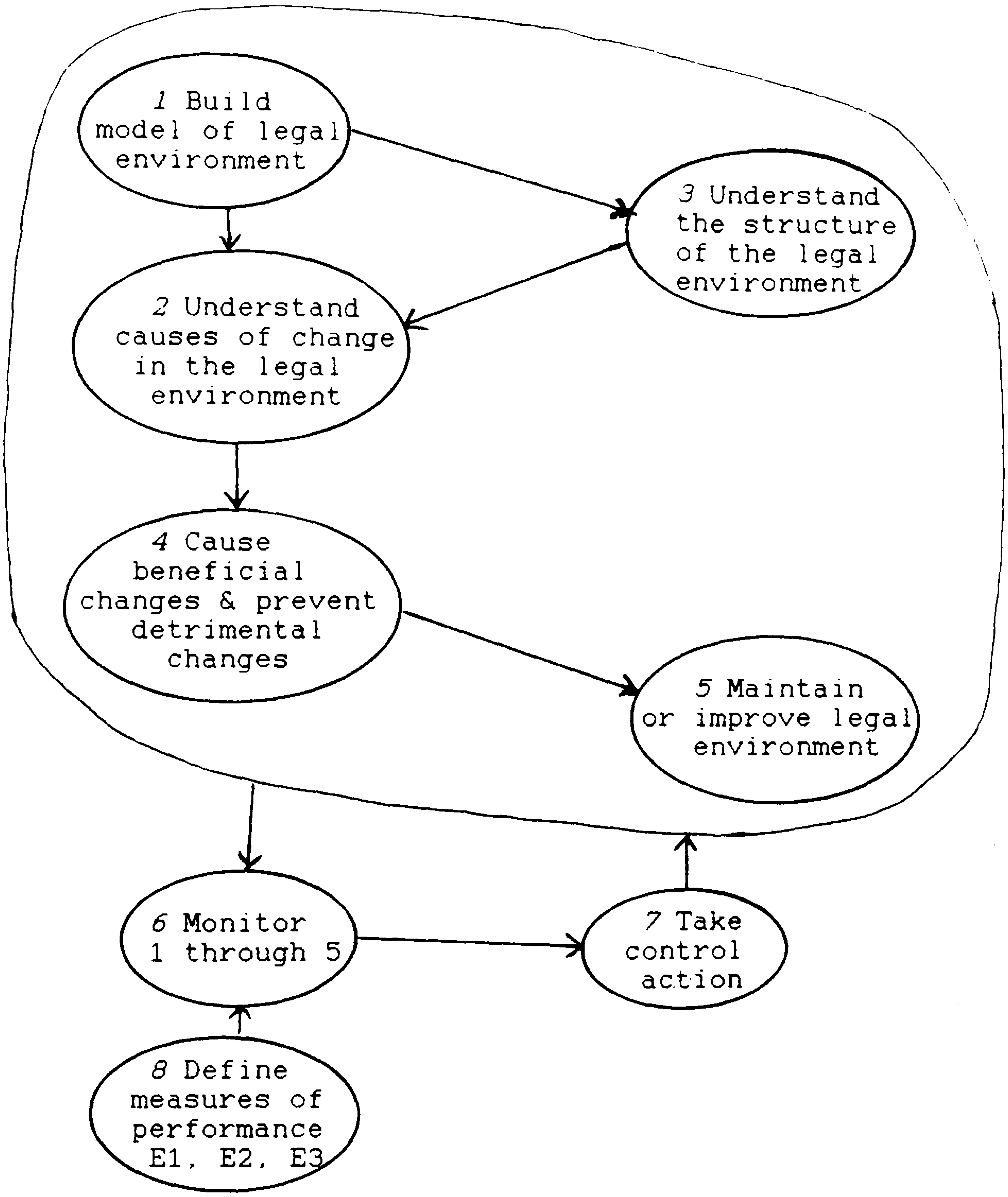


Figure 26

A system to improve or maintain the legal environment for product sales in the country.



E1: Legal environment improved or maintained
E2: Beneficial changes minus detrimental changes, divided by resources used.
E3: Increased product sales in the country

Figure 27

A system to build a model of the legal environment in the country. (Expansion of element 1, figure 26)

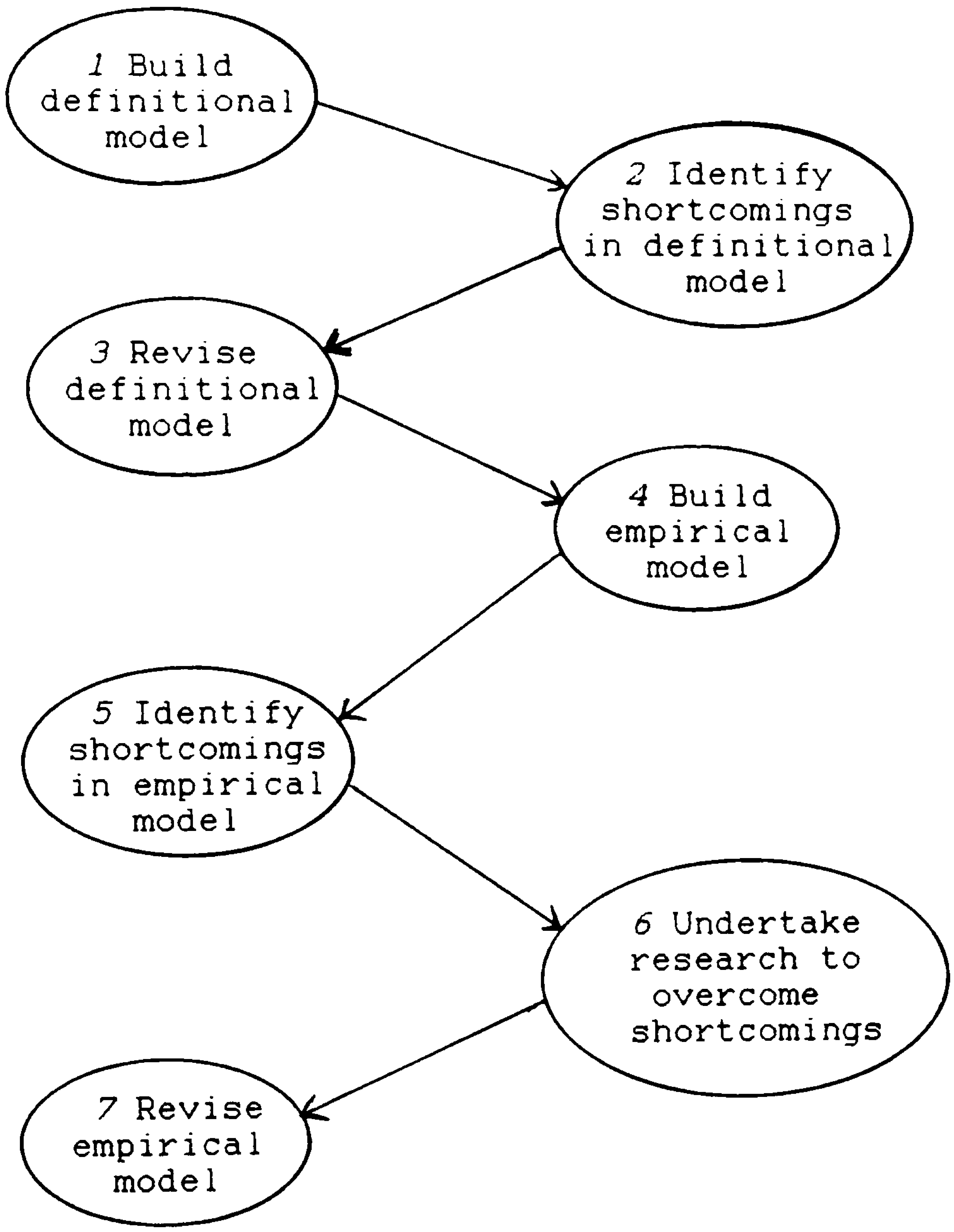


Figure 28

A system to prevent the introduction of new detrimental legislation and to repeal existing detrimental legislation.

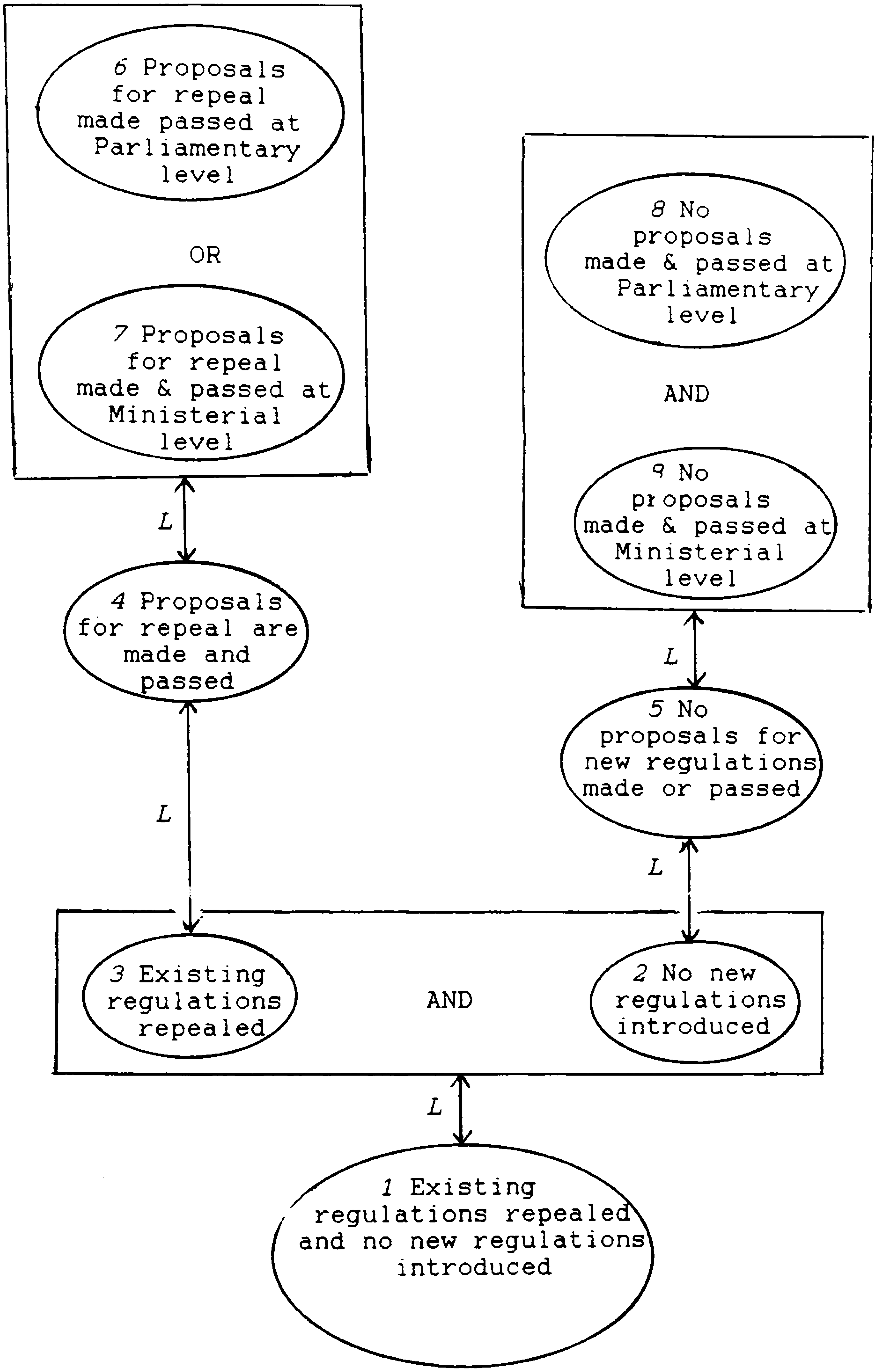


Figure 29

A system to have a proposal for the repeal of detrimental legislation made and passed at Ministerial level.

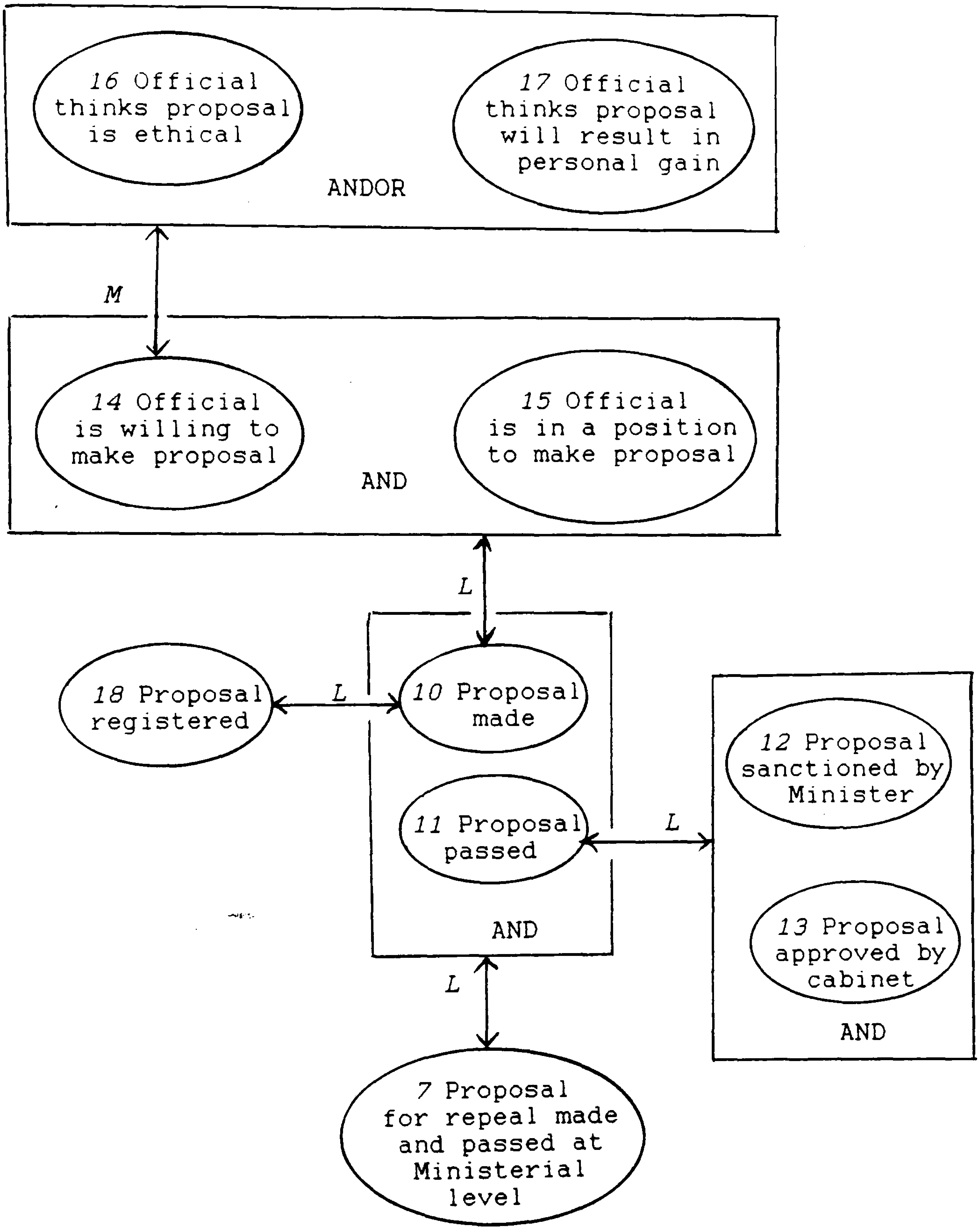


Figure 30

A system in which an official makes a proposal for personal reasons.

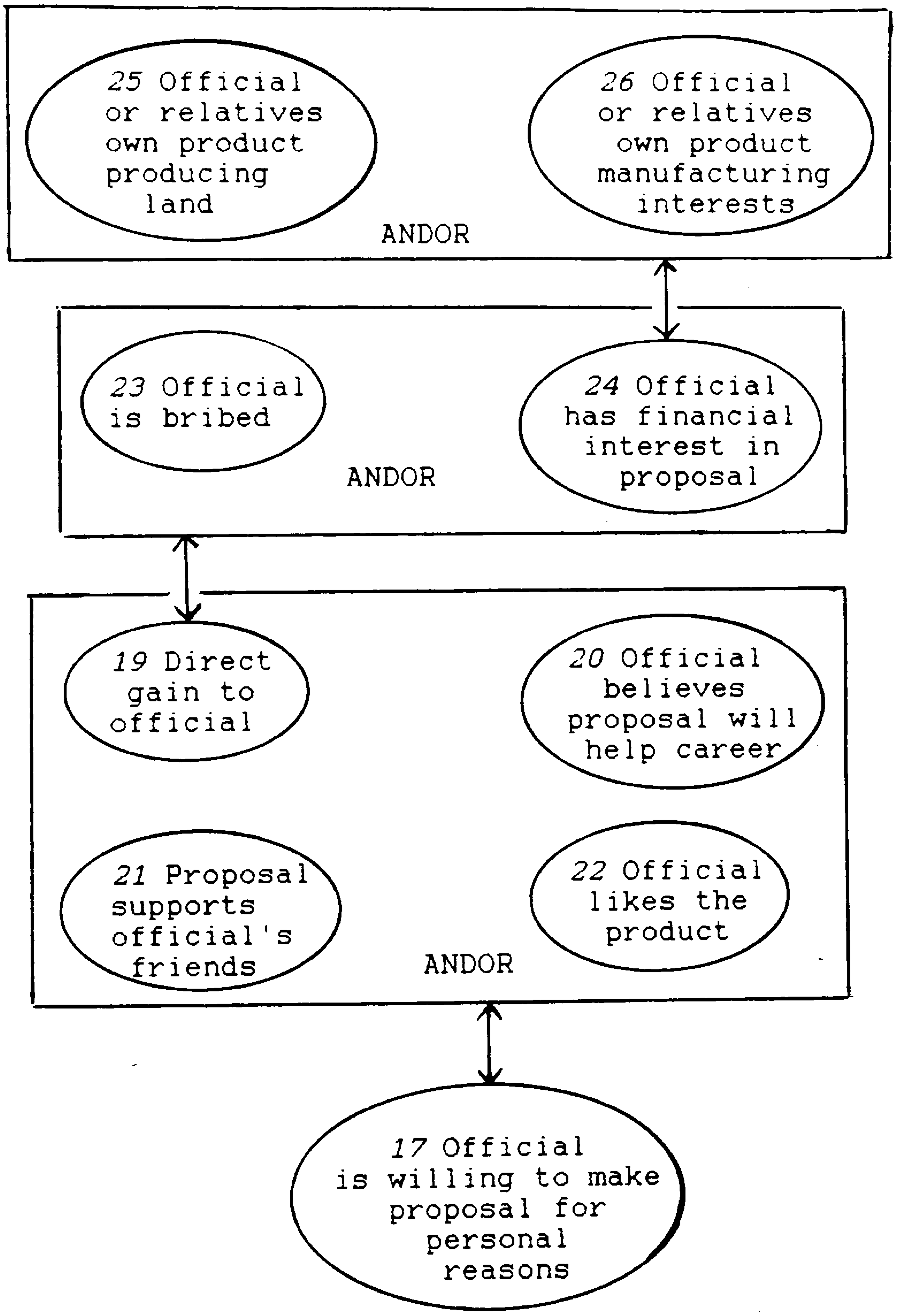


Figure 31

A system in which an official believes that proposing the repeal of regulations detrimental to product sales will be a career help.

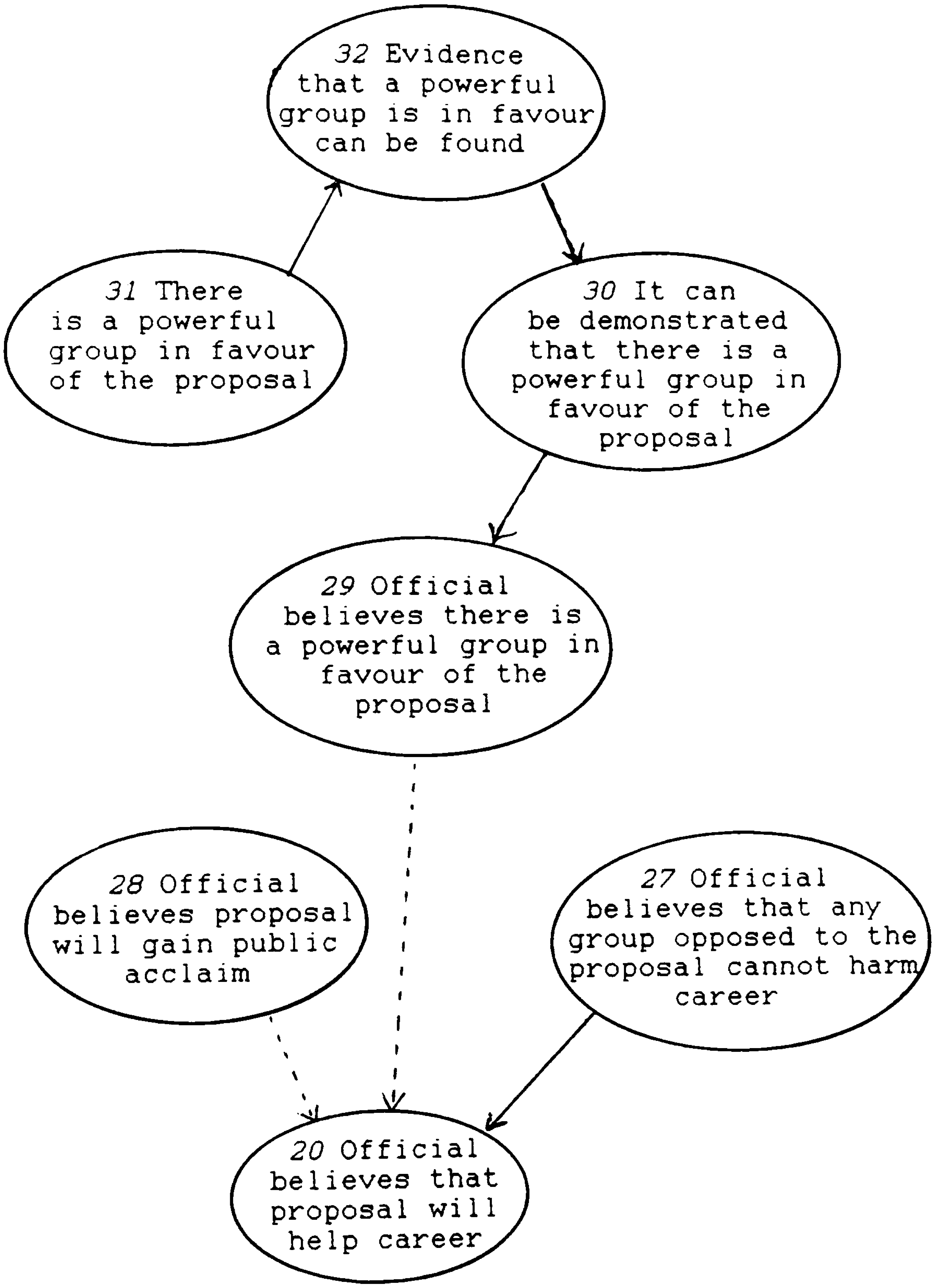


Figure 32

A system in which an official believes that proposing the repeal of regulations detrimental to sales will gain public acclaim.

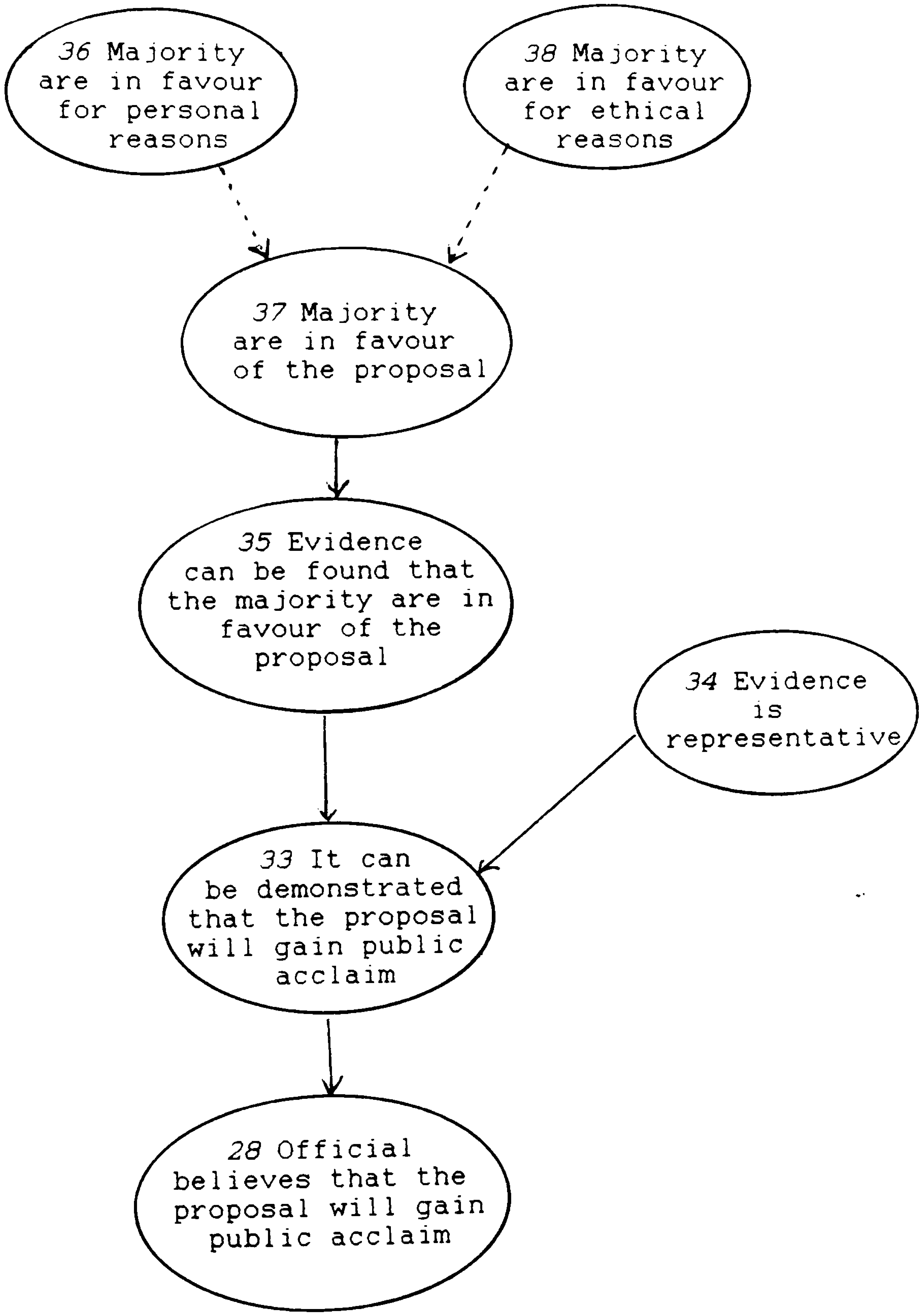


Figure 33

A system in which the majority of the public is in favour of a proposal to repeal regulations detrimental to product sales on ethical grounds.

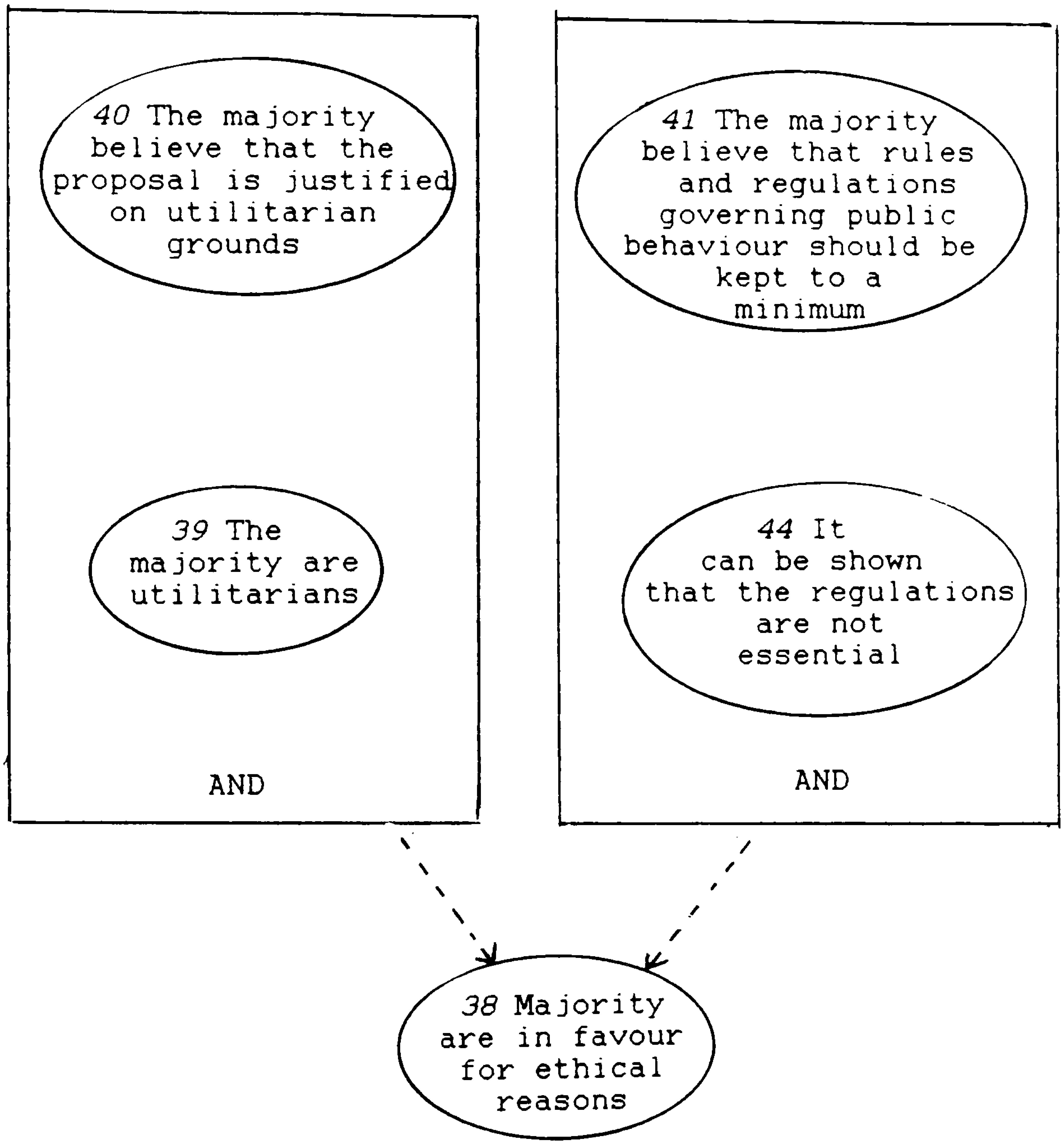


Figure 34

A system in which the majority of the public is in favour of a proposal to repeal regulations detrimental to product sales on utilitarian grounds.

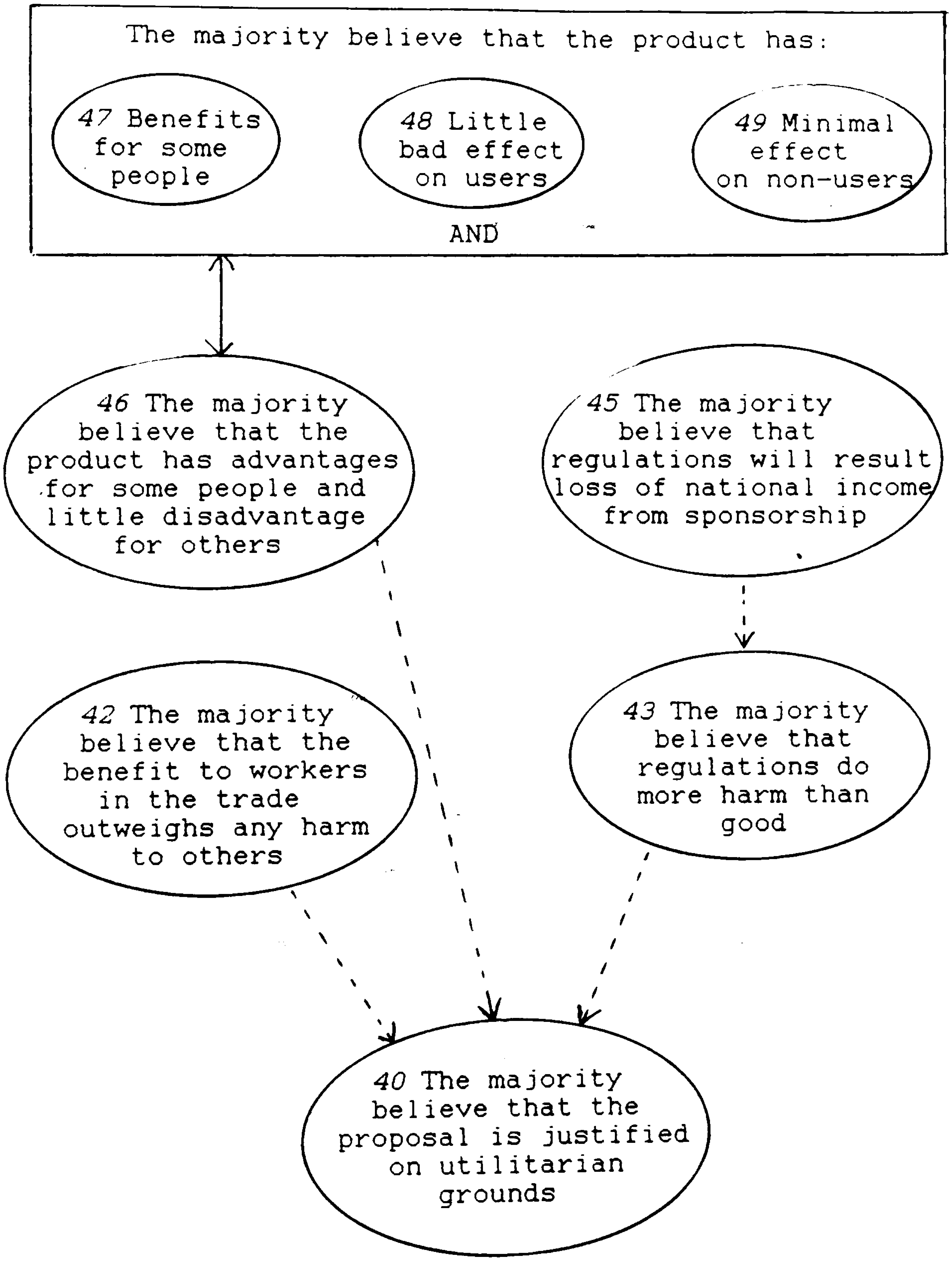


Figure 35

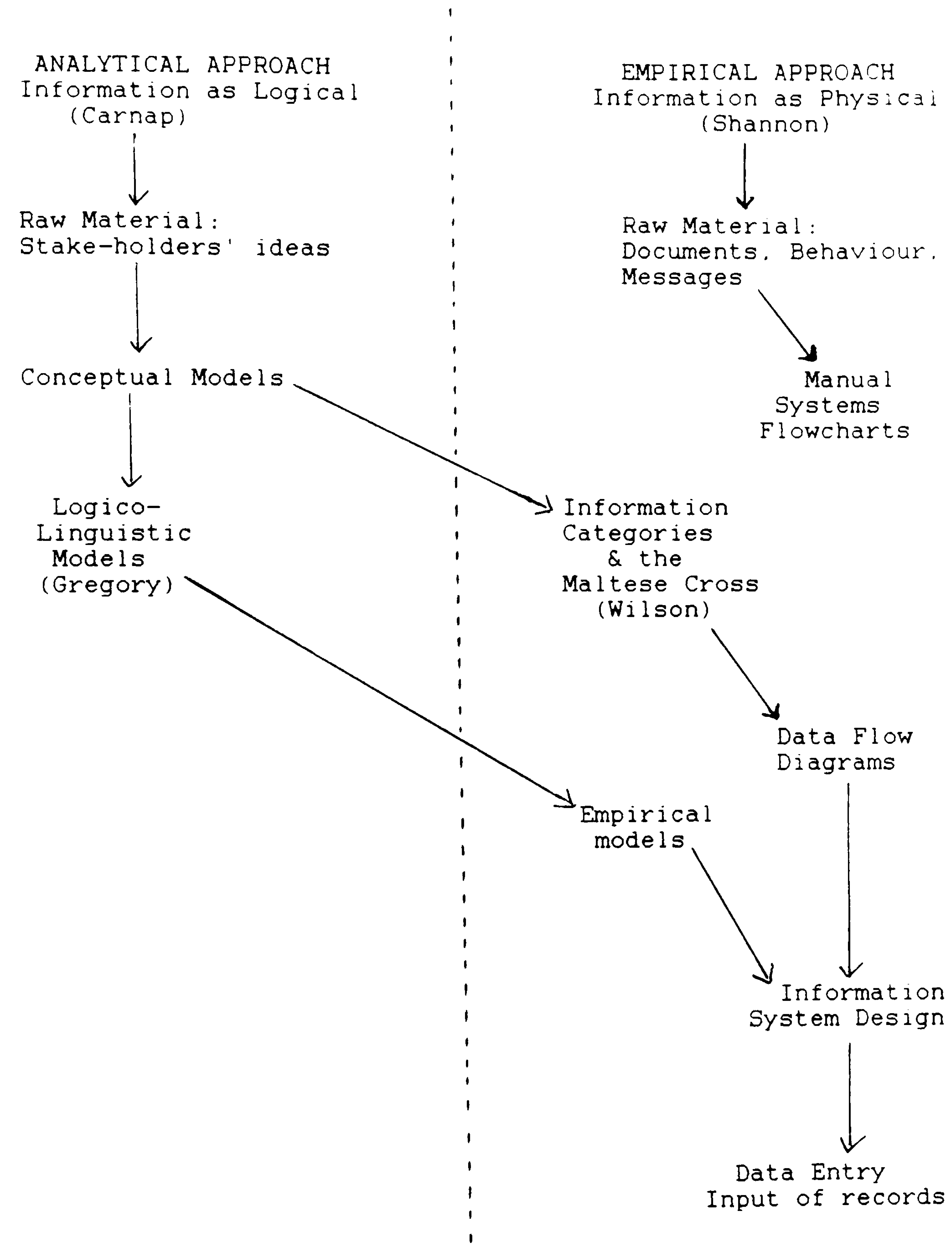

ANALYTICAL APPROACH
Information as Logical
(Carnap)
Raw Material:
Stake-holders' ideas
Conceptual Models
Logico-
Linguistic
Models
(Gregory)
EMPIRICAL APPROACH
Information as Physical
(Shannon)
Raw Material:
Documents, Behaviour,
Messages
Manual
Systems
Flowcharts
Information
Categories
& the
Maltese Cross
(Wilson)
Data Flow
Diagrams
Empirical
models
Information
System Design
Data Entry
Input of records

Table 1

Patient	Alpha	Antibody	Nose	Temp.	Eyes	Skin	Measles
ORIGINAL DATA							
Adam	yes	no	NA	NA	NA	NA	yes
Betty	yes	NA	NA	NA	NA	NA	NA
Colin	no	NA	runny	high	inflamed	NA	no
Dianna	NA	NA	runny	high	inflamed	rash	yes
Edward	no	NA	dry	high	inflamed	normal	no
Gertrude	yes	no	runny	high	inflamed	rash	yes
Harry	NA	NA	runny	high	inflamed	NA	NA
FIRST NEW							
Isabel	no	NA	runny	high	inflamed	rash	no
SECOND NEW							
Johnny	yes	no	dry	high	inflamed	rash	yes
THIRD NEW							
Karen	yes	no	dry	low	normal	normal	yes

13 BIBLIOGRAPHY AND REFERENCES

Auramaki, E., Lehtinen, E. & Lyytinen, K. (1988) A Speech-Act-Based Office Modeling Approach. *ACM Transactions on Office Information Systems,* Vol. 6, No.2.

Auramaki, E., Hirscheim, R. & Lyytinen, K. (forthcoming) Modelling Offices through Discourse Analysis: A Comparison and Evaluation of SAMPO with OSSAD and ICN. *The Computer Journal.*

Avison, D. E. & Wood-Harper, A. T. (1990) *Multiview: An Exploration in Information Systems Development,* Blackwell Scientific Publications, Oxford.

Ayer, A. J. (1936) *Language, Truth and Logic.* Victor Gollancz, London.

Ayer, A. J. (1969) *The Foundations of Empirical Knowledge.* Macmillan. London.

Bar-Hillel, Y. & Carnap, R. (1953) Semantic Information. *Brit. J. Phil. Sc.,* **4**, pp. 147-57.

Barlow, R. & Lambert, H. (1975) Introduction to fault tree analysis. in *Reliability and fault tree analysis* (R. Barlow Ed.). Society for Industrial and Applied Mathematics. Pennsylvania.

Carnap, R. (1950) *The Logical Foundations of Probability*. University of Chicago Press.

Checkland, P. B. (1981) Systems Thinking, Systems Practice. Wiley. Chichester.

Checkland, P. B. (1984) On the Meaning of Arrows in Conceptual Models. Internal Discussion Paper 2/84, (unpublished). Dept. of Systems. University of Lancaster.

Checkland, P. B. (1985) Achieving `Desirable and Feasible' Change: An Application of Soft Systems Methodology. *J. Opl Res. Soc.* **36**, 821 - 831.

Checkland, P. B. (1989) Soft Systems Methodology. In *Rational Analysis for a Problematic World.* (J. Rosenhead Ed.) John Wiley. Chichester.

Checkland, P. B. (1989b) An application of soft systems methodology, in Rosenhead, J. Rational Analysis for a Problematic World. Wiley. Chichester.

Checkland, P. B. (1992) Systems and Scholarship: The Need to do Better. *J. Opl Res. Soc.* Vol. 43. No. 11.

Checkland, P. B. & Scholes, J. (1990) *Soft Systems Methodology in Action.* John Wiley, Chichester.

Chellas, Brian F. (1980) Modal Logic: An Introduction. Cambridge University Press, Cambridge.

Churchman, C. West (1971) The Design of Inquiring Systems: Basic Concepts of Systems and Organisation. Basic Books. New York.

Clocksin, W. F. & Mellish, C. S. (1987) *Programming in Prolog*. Third Edition. Springer-Verlag. Berlin.

Codd, E. F. (1970) A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*. Vol. 13, No. 6.

Copi, I. (1968) *Introduction to Logic, Third Edition.* Macmillian. London.

Curtis, G. (1989) *Business Information Systems.* Addison-Wesley. Wokingham, England.

Davis, Lynda J. & Ledington, Paul W. (1991) Information in Action: Soft Systems Methodology. The Macmillian Press. London.

Forsyth, Richard (1984) Fuzzy reasoning systems. In Forsyth, Richard (Ed.) *Expert Systems: Principles and case studies.* Chapman and Hall. London.

Forsyth, Richard (1990) Ed. *Expert Systems: Principles and case studies. Second Edition.* Chapman and Hall. London.

Frege, G. (1892) Uber Sinn und Bedeutung. *Zeitschriff fur Philosophie und philosophische Kritik*, Vol 100.

Gettier, E. L. (1967) Is Justified True Belief Knowledge? in Phillips Griffiths, A. (Ed) *Knowledge and Belief.* Oxford University Press.

Grayling, A. C. (1990) *An Introduction to Philosophical Logic.* Duckworth, London.

Haack, S. (1978) *Philosophy of Logics.* Cambridge University Press. Cambridge.

Hanseth, O. (1991) Integrating information systems: the importance of contexts. in Stamper, P. *et al* (Ed.) *Collaborative Work, Social Communications and Information Systems.* North-Holland, Elsevier Science Publishers.

Hofstadter, D. R. (1980) *Godel, Escher, Bach: an Eternal Golden Braid.* Penguin Books, Harmondsworth.

Ishikawa, K. (1986) *Guide to Quality Control.* Second Edition. Asia Productivity Organization/Quality Resources, White Plains, New York.

Jayaratana, Nimal (1990) Systems Analysis: the Need for a Better Understanding. *International Journal of Information Management.* 10, 228-234.

Jones, C. & Clark, J. (1990) Effectiveness framework for supply chain management. *Computer-Integrated Manufacturing Systems,* **3 (4),** 196-206.

Kafka, Franz (1930) *The Castle.* Secker & Warburg.

Kant, Immanuel (1890) *Citique of Pure Reason.* (trans. Meiklejohn, J. M. D.) George Bell & Sons. London.

Kenny, Anthony (1973) *Wittgenstein.* Penguin Books. Harmondsworth.

Kripke, Saul A. (1982) *Wittgenstein on Rules and Private Languages.* Basil Blackwell, Oxford.

Kripke, Saul A. (1971) Identity and Necessity. In Munitz, Milton K. (Ed.) *Identity and Individuation.* New York University Press.

Lyytinen K. (1988) Stakeholders, Information Systems Failures and Soft Systems Methodology: An Assessment. *Journal of Applied Systems Analysis.* 15, pp. 61-81.

Mackie, J. (1980) *The Cement of the Universe: A study of Causation.* Clarendon Press. Oxford.

McDermott, D. & Doyle, J. (1980) Non-Monotonic Logic I. *Artificial Intelligence*. Vol. 13. pp. 41-72.

Merali, Y. (1992) Analytic Data Flow Diagrams: An Alternative to Physicalism. *Systemist* Vol. 14 (3), 190-198.

Merali, Y. (1993) Retaining Logical Consistency in Information Systems Development. *Proceedings of the Conference on the Theory, Use and Integrative Aspects of IS Methodologies*. British Computer Society Information System Methodologies Special Interest Group.

Mill, John Stuart (1843) *A System of Logic*. London.

Mingers, John (1990) The What/How Distinction and Conceptual Models: A Reappraisal. *Journal of Applied Systems Analysis*, 17, 21 - 28.

Minsky, M. (1975) A Framework for Representing Knowledge. In Winston (Ed.) *The Psychology of Computer Vision*. McGraw-Hill. New York.

Newton-Smith, W. H. (1990) Logic: An Introductory Course. Routledge. London.

Nogier, J. F. & Zock, M. (1992) Lexical choice as pattern matching. Knowledge Based Systems. Vol 5, No 3.

Papineau, D. (1978) *For Science in the Social Sciences.* Macmillian. London.

Popper, Karl R. (1963) Conjectures and Refutations. Routledge. London.

Patching, David (1990) Practical Soft Systems Analysis. Pitman. London.

Probert, S. K. (1991) A critical study of the National Computing Centre's Systems Analysis and Design Methodology, and Soft Systems Methodology. **M.Phil Thesis**, Newcastle Upon Tyne Polytechnic.

Probert, S. K. (1993) Logic and conceptual modelling in Soft Systems Methodologies. *Proceedings of the Conference on the Theory, Use and Integrative Aspects of IS Methodologies.* British Computer Society Information System Methodologies Special Interest Group.

Quine, Willard Van Orman (1961) *From a logical point of view.* Harvard University Press. Cambridge, Massachusetts.

Ringland, G. A. & Duce, D. A. (1988) *Approaches to Knowledge Representation: An Introduction.* Research Studies Press, Taunton. John Wiley, New York.

Robinson, R. (1954) *Definition.* Clarendon Press, Oxford.

Rosenhead, J. (1989) Introduction: old and new paradigms of analysis. In *Rational Analysis for a Problematic World.* (J. Rosenhead Ed.) John Wiley. Chichester.

Russell, B. (1905) On Denoting. in Russell (1956)

Russell, B. & Whitehead, A. N. (1910) Principia Mathematica. Cambridge University Press.

Russell, B. (1918) The Philosophy of Logical Atomism. in Russell (1956).

Russell, B. (1924) Logical Atomism. in Russell (1956)

Russell, B. (1956) Marsh, R. Ed. *Logic and Knowledge.* Allen & Unwin. London.

Shannon, C. E. (1948) A Mathematical Theory of Communication. *Bell System Technical Journal.* Vol. 27 (July) pp. 379-423, (October) pp. 623-56.

Shannon, C. E. & Weaver, W. (1949) The Mathematical Theory of Communication. University of Illinois Press, Urbana.

Sowa, John F. (1992) Logical structures in the lexicon. Knowledge Based Systems. Vol 5, No 3.

Sowa, John F. (1984) Conceptual Structures: Information Processing in Mind and Machine. Addison Wesley. Reading, Massachusetts.

Stamper, R. (1973) *Information in Business and Administrative Systems.* Batsford, London.

Stamper, R. (1987) Semantics. In Boland, R. J. and Hirschheim, R. A. Eds. *Critical Issues in Information System Systems Research.* Wiley, Chichester.

Stephens, R. A. & Wood, J. R. G. (1991) Information Systems as Linguistic Systems: A Constructivist Perspective. In Jackson *et al* Eds. *Systems Thinking in Europe.* Plenum Press. New York & London.

Stewart, Ian (1993) Children of Invention. Times Higher Education Supplement. March 19.

Stove, D. (1965) Hume, Probability, and Induction. Philosophical Review. Vol. LXXIV. pp. 160-177.

Suppes, P. (1970) A Probalistic Theory of Causality. Amsterdam.

Taylor, R. (1963) Causation. *The Monist,* **47 (2)**, 287-313.

Turing, Alan M. (1950) Computing Machinery and Intelligence. *Mind* LIX, no. 2236.

Wilson, Brian (1984) *Systems: Concepts, Methodologies and Applications,* John Wiley, Chichester.

Wilson, Brian (1990) *Systems: Concepts, Methodologies and Applications, Second Edition,* John Wiley, Chichester.

Wittgenstein, L. (1922) *Tractatus Logico-Philosophicus.* Ogden & Richards translation, Routledge & Kegan Paul.

Wittgenstein, L. (1953) *Philosophical Investigations.* Blackwell, Oxford.