

# **Belief Ascription and Model Generative Reasoning: joining two paradigms to a robust parser of messages.**

*Yorick Wilks and Roger Hartley*  
Computing Research Laboratory  
New Mexico State University  
Box 30001, Las Cruces, NM 88003.

## **ABSTRACT**

This paper discusses the extension of *ViewGen*, a program for belief ascription, to the area of intensional object identification with applications to battle environments, and its combination in a overall system with MGR, a Model-Generative Reasoning system, and PREMO a semantics-based parser for robust parsing of noisy message data.

*ViewGen* represents the beliefs of agents as explicit, partitioned proposition-sets known as environments. Environments are convenient, even essential, for addressing important pragmatic issues of reasoning. The paper concentrates on showing that the transfer of information in intensional object identification and belief ascription itself can both be seen as different manifestations of a single environment-amalgamation process. The entities we shall be concerned with will be ones, for example, the system itself believes to be separate entities while it is computing the beliefs and reasoning of a hostile agent that believes them to be the same entity (e.g. we believe enemy radar shows two of our ships to be the same ship, or vice-versa. The KAL disaster should bring the right kind of scenario to mind). The representational issue we address is how to represent that fictional single entity in the belief space of the other agent, and what content it should have given that it is an amalgamation of two real entities.

A major feature of the paper is our work on embedding within the *ViewGen* belief-and-point-of-view system the knowledge representation system of our MGR reasoner, and then bringing together the multiple viewpoints offered by *ViewGen* with the multiple representations of MGR. The fusing of these techniques, we believe, offers a very strong system for extracting message gists from texts and reasoning about them.

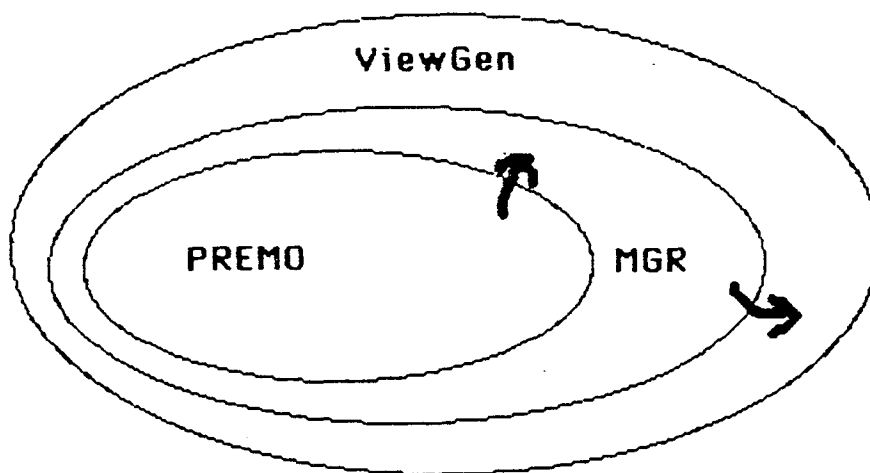
## **INTRODUCTION**

The purpose of this paper is to bring together the reasoning techniques used in the MGR and *ViewGen* systems and combine those with the PREMO robust parsing system (see references in separate sections below) to yield a combined approach to understanding text, particularly the interpretation or gisting of text which may be ill-formed, fragmented, stylized or even self-contradictory (when produced under stressful conditions, but evaluable against an appropriate knowledge base nonetheless). We also aim to build in pointers to the evaluation of the system as we go. Another guiding principle is to combine if possible, the benefits of being knowledge and situation based, when needed for interpretation, with the benefits that come from large scale automated analysis techniques drawn from the use of machine readable dictionaries and large text samples.

We shall direct our examples where possible to realistic military scenarios drawn from other Army-sponsored work at CRL and construct scenarios based on ship or troop unit sightings and casualty reports. However, our techniques are better, we hope, than the individual scenarios we can construct. In other cases we shall use the standard cases with which the systems have been developed.

The overall aim is to produce a system that takes in noisy messages and data, has access to both Blue data-bases and military doctrine (in terms of beliefs and goals) and, in certain circumstances, access also to Red Army beliefs and goals. A number of possible scenarios are tentatively put forward at the end of the paper, in which the combination of the techniques discussed here could provide advice to, and a check upon the decisions of, the G2 officer in a Blue Army Corps, particularly in terms of the consistency of the advice offered the commander and consilience of the belief and data structures maintained at other lower levels of the system (e.g. Divisions).

In the first three sections of the paper we summarize the PREMIO, MGR and ViewGen techniques and then proceed to a discussion of how to link them: in general we envisage an overall system in which information moves (outward in the diagram below) from the robust parser PREMIO, to the MGR, the generator of alternative scenarios/models, to ViewGen the point-of-view shell that considers the possible points of view in terms of the environments of other agents (e.g. Blue Generals view of Red General's view of a particular tank division). Both MGR and ViewGen also act as filters of possible but inapplicable models of the situation.



### **PREMIO: A ROBUST PARSER OF MESSAGES**

**PREMIO:** the **PRE**ference Machine Organization is a knowledge-based Preference Semantics parser (Wilks 1972, 1975, 1978; Boguraev 1979; Carter 1984, 1987; Fass 1986, 1987, 1988; Huang 1984, 1988; Slator 1988a, 1988c) due to Brian Slator, with access to the large, text-specific, lexical semantic knowledge base created by the lexicon-provider of the CRL project on large scale lexical extraction from machine readable dictionaries (Wilks et al. in press). A fuller description of the relationship of PREMIO to that project appears in (Slator & Wilks 1989). PREMIO is the parser we intend to use initially for initial processing of military reports (sightings, casualties etc.) that have been generated rapidly under adverse

conditions.

Preference Semantics is a theory of language in which the meaning for a text is represented by a complex semantic structure that is built up out of smaller semantic components; this compositionality is a fairly typical feature of semantic theories. The principal difference between Preference Semantics and other semantic theories is in the explicit and computational accounting of ambiguous, metaphorical, and linguistically non-standard language use; which is to say, it is intended to deal with the actual world of English texts outside linguistic studies, ranging from newspaper texts (well-formed but full of the non-standard phenomena just listed) to rapidly written diagnostic reports, like machine-repair, or reports of sightings, which have all those features and are grammatically ill-formed in addition..

The links between the components of the semantic structures are created on the basis of semantic preference and coherence. In text and discourse theory, coherence is generally taken to refer to the meaningfulness of text. Fass (1987) suggests that in NLP work such as Preference Semantics the notions of "satisfaction" and "violation" (of selection restrictions or preferences) and the notion of "semantic distance" (across structured type hierarchies) are different ways of characterising the meaningfulness of text; they capture different coherence relations. The original systems of Preference Semantics (Wilks 1972, 1975, 1978), were principally based on the coherence relation of "inclusion" (semantic preferences and selection restrictions); the emphasis in PREMO is more on the coherence relation based on semantic distance, although the original notions of coherence also survive.

In Preference Semantics the semantic representation computed for a text is the one having the most semantically *dense* structure among the competing "readings." Semantic density is a property of structures that have preferences regarding their own constituents, and satisfied preferences create density. Density is compared in terms of the existence of preference-matching features, the lack of preference-breaking features, and the length of the inference chains needed to justify each sense selection and constituent attachment decision. The job of a Preference Semantics parser, then, is to consider the various competing interpretations, of which there may be many, and to choose among them by finding the one that is the most semantically dense, and hence preferred.

PREMO is a robust system for parsing natural language organized along the lines of an operating system. The state of every partial parse is captured in a "process control block" structure called a *language object*, and the control structure of the preference machine is a priority queue of these language objects. The language object at the front of the queue has the highest score as computed by a preference metric that weighs grammatical predictions, semantic type matching, and pragmatic coherence. The highest priority language object is the intermediate reading that is currently most preferred (the others are still "alive," but not actively pursued); in this way the preference machine avoids combinatorial explosion by following a "best-first" strategy for parsing. Each "ready" process in the system captures the state of a partial parse with priority given to each parse "process" on the basis of a preference semantics evaluation. The "time-slice" for each process is whatever is needed to move forward one word in a local process sentence buffer (where each process operates on a private copy of the current sentence). After every time slice, the preference/priority for the currently "running" parse is re-computed and the language object for that process is returned to the priority queue. The first process to emerge from the queue with its sentence buffer empty is declared the winner and saved. This strategy is both a run-time optimization and an application of the "Least Effort Principle" of intuitively plausible language processing. The parsing is robust in that some structure is returned for every input, no matter how ill-formed or "garden-pathological" it is.

A principal advance in PREMO over earlier Preference Semantics work is that it overcomes the semantic localism of that work that allowed peculiar but telling counter-examples to be created. A better-known one is due to Phil Hayes (quoted in Boden 1977): "He licked the gun all over and the stock tasted good". A Preference Semantics system might well get "stock" resolved to "soup" rather than "gun stock" based on the local preferences of the last clause, because it had no overall GUN context, or indeed any way of expressing that. The frame movement of the Seventies was intended to provide such contextual theories (e.g. Minsky 1975) but it has not been generally accepted that it did so. The script-based parsers from Yale (e.g. Gershman, 1977) were created in the same topic-based spirit tended to fall into the opposite fault: of not understanding what was said unless it did wholly conform to expectations.

PREMO is intended to, and does, meet both requirements: the bottom-up coherence of preference combined by weightings with the topic-subject codes provided for LDOCE (Longmans Dictionary of Contemporary English, Procter et al.1978) within the lexical acquisition project at CRL mentioned earlier.

### **MGR: MODEL GENERATIVE REASONING**

The global objective of this project is to investigate general mechanisms for symbolic problem solving in task environments where data are noisy and where the problems addressed require objects to be related in ways unanticipated by the designer. This objective is motivated by the frequently reported result that problem solvers developed using knowledge-based programming techniques are brittle when there is significant noise or novelty in the task environment. (Coombs & Hartley, 1987, 1988a,b, Fields et al. 1988a,b).

Typical AI problem solvers employ fixed semantic relations (*e.g.* rules) to search for solutions. While the structure of these relations plays a significant part in the effectiveness of a problem solver, the manipulation of structure is not currently part of AI problem solving methodology. However, in the absence of reliable data or complete domain knowledge, *i.e.* when rules are unreliable, the problem solver will have to rely on the structural characteristics of interactions between data and knowledge for control. Problem solvers will, therefore, have to be sensitive to structural as well as semantic relations. MGR is a problem-solving architecture capable of this sensitivity.

The basic MGR approach to reasoning is formally related to generalized set covering (GSC), which has been used effectively in automated problem solving for tasks where success is highly dependent on the quality of hypotheses (*e.g.*, medical diagnosis). MGR extends GSC to its full generality by representing problem solving objects in terms of graphs and using the mathematical theory of graphs for the specification of problem solving procedures. Graph theory also gives us an ideal vehicle for the exploration of structural relations.

Starting with general set covering as the foundation model, we propose to: i. expand the basic set covering operations to their full generality in terms of graph theory to produce a structural formulation of covering operations, ii. relate the graph theoretic descriptions of set covering operations to MGR operations, and iii. prove consistency of the graph theoretic formulation of MGR and investigate the degree of its completeness. Following this we will investigate strategies for coping with noise and novelty in terms of manipulating relations between the graphs that represent problem data and knowledge. This will require the implementation of an explicitly graph theoretic version of the MGR problem solving architecture, with the capability of computing the structural characteristics of its computational objects and for using these as control parameters. These will then be used to address issues of tractability arising from problem solving within noisy and novel task environments through a series of experiments on (i) countering the effects of noise in data, and (ii) enabling a system to adapt to novel relations between objects.

#### **The Abductive Mechanism in MGR**

Mechanizing any inference procedure runs the risk of being tied down to syntactic issues, whereas the real problems lie with the semantics of the techniques. Our technique is actually far more semantic in nature than the logic-based approaches of Charniak, Hobbs and the sort of approach advocated by Levesque (Charniak, 86, Hobbs, 89, Levesque, 89). In his paper, Levesque distinguishes the set-covering ideas of Reggia and Allemang *et al.* (Nau and Reggia, 86, Allemang et al, 87) from the logic-based methods. For us the issue rests on the nature of an *explanation* and the utility of such a structure, were it to be available. A logical approach would call anything an explanation as long as the facts being explained are entailed by the explanation and some body of assumed knowledge. Schematically this is:

If there is a set of facts  $F$  and a body of domain knowledge  $D$ ,  
then a formula  $E$  explains  $F$  iff

$$E \wedge D \supset F$$

Mechanizing this involves computing the best possible  $E$ . This is rather like computing  $x$  in the simple equation  $3 + x = 5$ . Calling '2' an explanation of '5' in the context of '3' and the whole of arithmetic is what this approach would tend to say. As Levesque points out there are a lot of assumptions built into this approach, not the least being that the knowledge in  $D$  must be complete (in some non-trivial sense) and consistent. For any problem solver that claims generality, these are hard requirements to meet. The set covering approach, however, relaxes the stringent requirements of a logic-based system. Instead of requiring a consistent knowledge base, we only require that there be coherent subsets. Coherence in MGR rests on the idea of obeying selectional constraints within a type hierarchy. The abductive mechanism then finds these subsets by insisting that every term in the facts being explained is covered by at least one schema in the knowledge base (we could call these rules, but this would mislead, since covering does not impose a preferred form on the components of the knowledge base). Cover can be computed according to parsimony (the best explanation is the simplest) or according to any other suitable criterion. Logical entailment also uses parsimony to produce the simplest expressions. The set covering idea is best expressed schematically as:

Given a set of terms  $T$ , a set of facts  $F$  and a set of schemata  $D$ ,  
each of which contain expressions of these terms, an explanation  $E$   
is said to exist iff:

$$t(E) = \cup t(D') \text{ where } D' \subset D, \text{ and} \\ t(F) \subseteq t(E)$$

The computation of  $E$  depends on finding the smallest subset  $D'$  that contains all of the terms in  $F$ . Since the relationship between the terms and the components of  $D$  that contain them is pre-computed in MGR, this resolves into an issue of minimizing the Boolean formula that is the conjunction of all the disjunctions each formed by covering one term in  $F$  with the schemata that contain it. We are also currently looking at ways to achieve the same result with less than parsimonious covers to be used when parsimony is too strong a principle. The resultant explanations (the models in MGR) are coherent models of some aspect of real-world behavior, not merely loose connections of atomic formulae as they would be in a logic-based system. They can then be used for prediction (using simulation-like techniques, or deduction if you prefer) with real data. Far from being merely syntactic, the models in MGR bear more relationship to the models typically produced by a simulation language, rather than the structures employed by, say the ATMS of de Kleer (Reiter and de Kleer, 87, Levesque, op.cit.) The successes of set covering in (Nau and Reggia, op. cit.) and (Josephson, 87) are a pointer to the approach work supporting of MGR.

### Formalization of the MGR Symbolic Problem Solver

MGR employs four operators: *Generalize Gn*, *Merge Mr*, *Specialize Sp*, and *Fragment Fr*. These operators are implemented using the intensional **join** and **project** operators as defined by Sowa (*op. cit.*) as analogous with the extensional relational data-base operations join and project. The operators are defined as mappings over the sets  $F$  (*facts*),  $D$ , (*definitions*) and  $M$  (*models*) as follows:

$$\begin{array}{ll} \mathbf{Gn}: \text{PS}(M) \rightarrow M & \mathbf{Mr}: \text{PS}(M) \rightarrow M \\ \mathbf{Sp}: M \times \text{PS}(D) \rightarrow \text{PS}(M) & \mathbf{Fr}: M \times \text{PS}(F) \rightarrow \text{PS}(M) \end{array}$$

The notation  $\text{PS}(X)$  denotes the power set of the set  $X$ .

The MGR operators are implemented as follows. **Gn** takes as input a subset  $M'$  of  $M$ , and returns the **project** of the models in  $M'$ . **Mr** takes as input a subset  $M'$  of  $M$ , and returns the **join** of the models in

$M'$ . **Sp** takes as input a subset  $D'$  of  $D$  and a model  $m \in M$ , and returns a set  $M'$  of models, each of which is a **join** of  $m$  with all of the elements of  $D'$  with which it is joinable. **Fr** takes as input a subset  $F'$  of  $F$  and a model  $m \in M$ . **Fr** then computes the **join**  $f$  of the elements of  $F'$ . If  $m$  has a set of disconnected subgraphs  $g_i \dots g_k$  such that  $\text{project}(g_i \dots g_k) = f$ , then **Fr** returns  $\{g_i, \dots, g_k\}$ .

Informally, **Sp** takes a model as input, and generates a set of larger, more specialized models by adding definitions. The role of **Sp** is, therefore, to “glue” knowledge to facts to create models that cover the facts. **Fr** opposes **Sp** by breaking models into fragments in a way that preserves the information contained in facts, but may destroy information obtained from definitions. **Fr** thus generalizes definitional information, but not factual information. The role of **Fr** is to break apart models that do not cohere with all of the available the facts in order to generate fragments that can be recombined. **Gn** and **Mr** take subsets of models as input, and generate single models as output which are, respectively, less or more specialized than the models from which they were generated. **Gn** is capable of generalizing both factual and definitional information; its role is to maintain coherence with the facts by removing over-specializations. **Mr** merges models whenever possible; its role is to generate models that have the greatest possible covering power. All of the operators write over the model population; the set of models available for operating on, therefore, changes on every cycle.

### Current Work on MGR

1. The techniques of model generative reasoning are being applied to situation analysis for TRADOC through the Army Intelligence School, at Fort Huachuca. MGR is being used to develop hypothetical models of enemy intentions from an input of intelligence reports presented in terms of a set of ‘leading indicators’. Stored knowledge consists of schematic definitions of types of maneuver (both static and dynamic components) typically carried out by the enemy army.
2. A formal analysis of the MGR operators is proceeding, leading towards a better understanding of their algorithmic complexity. Eventually this will lead to a parallel version of the MGR architecture. This work is supported by Harry Diamond Labs. through the ACT program.
3. Work has progressed in characterizing the management of alternative hypotheses as generated by MGR through dynamic systems control techniques. We have also looked at the genetic algorithm as a variant that could be used to control MGR’s search for the ‘best’ model.

## ViewGen: A POINT OF VIEW SHELL FOR REASONING ABOUT BELIEFS

### Introduction

An AI system that takes part in discourse with other agents must be able to reason about the beliefs, intentions, desires, and other propositional attitudes of those agents, and of agents referred to in the discourse. This is especially so in those common situations when the agents’ beliefs differ from the system’s own. Thus, the question of how to represent and reason about propositional attitudes is central to the study of discourse.

Clearly, this question is really about the beliefs, and so forth, that the system *ascribes* to the agents, on the evidence presented by the discourse itself and by context and prior information, since persons have no direct access to each others’ mental states. We view the ascription problem as being a fundamental one. It has been the focus of our past work on propositional attitudes and that of our immediate colleagues Afzal Ballim and John Barnden: (Ballim, 1986, 1987, 1988; Ballim & Wilks, forthcoming; Barnden, 1983, 1986a, 1986b, 1987a,b, 1988a,b, to appear; Wilks & Ballim, 1987, 1988, in press (a,b); Wilks & Bien, 1979, 1983). Ascriptional reasoning is profoundly dependent on the communicative context, general information that the system has about the world, and special information the system has about the agents at hand. Moreover, there are major pragmatic features of discourse, such as *speech acts*, metaphor, and the determination of the *intensional entities* in play in a discourse, that any system for ascribing beliefs to agents must address. We would go further, and assert that even the most apparently superficial aspects of natural language understanding depend on belief

ascription: such as propositional phrase attachment. Anyone hearing a sentence with the all-too-familiar structure:

*He told his mother about the murder in the park.*

will interpret it differently according to whether he believes the speaker believes there was a murder in a park and that the speaker believes the hearer believes that too. The function of our basic program *ViewGen* is to create, or as we shall call it, *ascribe*, environments into which appropriate beliefs can be segregated so that parsing and reasoning can be done in that limited environment.

We have described the basic algorithm in *ViewGen* in the publications above, and we address the issue of basic parsing issues seen a belief phenomena elsewhere. Here our purpose is simply to review the basic ascription mechanism and then show its extension to the phenomena such as the identification of intensional objects.

In interpreting an utterance by an agent, the system must ascribe a speech act to that agent; and doing that is a matter of ascribing specific intentions, beliefs, desires, expectations and so on to the agent. Thus, speech act ascription is an important special case of ascriptional reasoning. That speech-act considerations make reasoning about propositional attitudes essential for the computational modelling of discourse has been established at least since the work of Perrault and his colleagues (e.g. Perrault & Allen, 1980). A major difference between that work and ours is that they took the content of belief environments to be already established, whereas our approach is based on the real-time computation of the contents of such belief environments.

The work of Maida (1984, 1986) shares many of the concerns of the current work: his diagrammatic representations of nested beliefs are isomorphic to those of Wilks & Bien (1979) and Shadbolt (1983). His concerns are with the problem of shared reasoning strategies between believers and how, for example, you could establish that a dialogue partner also used *modus ponens*. We argue, on the contrary, that this phenomenon is best handled by general default assumptions, as are the concrete contents of belief. No finite set of dialogue observations ever could establish conclusively that another believer was using *modus ponens*. That being so, concentration on such issues that are not susceptible to proof seems to us only to delay the central issue, which is how to infer heuristically the actual beliefs of other believers. Maida (1983) is also concerned with the very important, and we believe quite separable issue, of a heuristic rule for identifying intensional individuals under different descriptions. Konolige's (1983) work has strong similarities to that just noted; he considers what he calls views, for which he writes e.g.,  $v=John,Sue, Kim$ , to mean *John's view of Sue's view of Kim's beliefs*. But he has no effective construction for the content of such views. Rather, he is concerned with giving an account of limited deduction in such views, an important process, but not relevant to issues of constructing individuals' views. Dinsmore (1987) has been concerned with what he terms the "algebra of belief spaces" but, although the term is highly general, the focus of his attention is always in fact the notions of presupposition and counterfactuals, which are not notions we treat explicitly here, and his treatment of them may well be compatible with our own general approach.

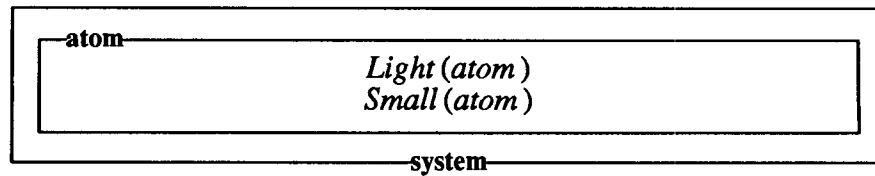
### ***ViewGen*: The basic belief engine**

A computational model of belief ascription is described in detail elsewhere (Wilks & Bien, 1979, 1983; Ballim, 1987; Wilks & Ballim, 1987; Ballim & Wilks, forthcoming) and is embodied in a program called *ViewGen*, two versions of which have been written in prolog by Afzal Ballim. The basic algorithm of this model uses the notion of default reasoning to ascribe beliefs to other agents unless there is evidence to prevent the ascription. Perrault (1987, forthcoming) and Cohen & Levesque (1985) have also recently explored a belief and speech act logic based on a single explicit default axiom. As our previous work has shown for some years, the default ascription is basically correct, but the phenomena are more complex (see below) than are normally captured by an axiomatic approach.

*ViewGen*'s belief space is divided into a number of topic specific partitions (topic environments). These environments can be thought of as a less permanent version of *frames* (Minsky, 1975; Charniak, 1978) or more suitably in terms of Wilks (1978) as pseudo-texts (henceforth PTs). In effect, a PT is a set of unsorted, unrefined items of knowledge. These PTs are general items and are not only stored for individual human beings, but also for groups of humans, objects, and abstract ideas. Their hierarchical and inheritance relations are discussed in Wilks (1978) and Ballim & Wilks (forthcoming). We justify the general notion of explicit environment in the next section.

*ViewGen* is a program that generates a type of environment known as a *viewpoint*. A viewpoint is some person's beliefs about a topic. Within *ViewGen*, all beliefs are ultimately beliefs held by the system (e.g., the system's beliefs about France, what the system believes John believes about cars, etc.) and so, trivially, lie within the system's viewpoint. The

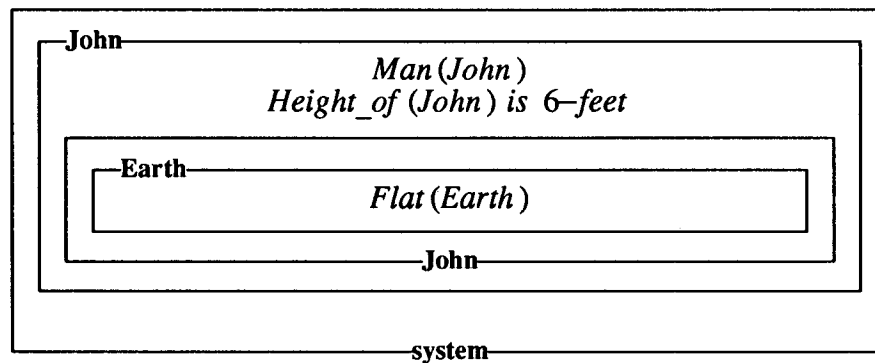
system's view of some topic (say, atoms) is pictorially represented as:



This diagram contains two types of environments: First, there is the box labelled with "system" at the bottom. This is a "believer environment" or "viewpoint." Viewpoints contain "topic environments," such as the box labelled with "atom" at the top of it. A topic environment contains a group of propositions about the "topic." So, for example, the above diagram conveys that the system believes that atoms are light and small. Topic boxes are motivated by concerns of limited reasoning (see Section 5.1 on relevance, and also Wilks & Bien, 1983). In short, it is envisaged that reasoning takes place "within" a topic environment, as if it were the environment of a procedure in a programming language.

Within *ViewGen*, environments are dynamically created and altered. *ViewGen*'s "knowledge-base" can be seen as one large viewpoint containing a large number of topic environments, with each topic environment containing a group of "beliefs" that the system holds about the topic. The reader should note that each proposition in a topic environment has at least one symbol identical to the name of the topic. Each such proposition is therefore *explicitly* about the topic. There are, however, implicit ways in which a proposition can be "about" (or "relevant to") a topic. The simplest cases are generated by inheritance in the usual way: for example, if John is a man, then any proposition in a "man" topic environment is implicitly or indirectly about John. However, we choose not to put such a proposition in the John topic box, and will justify that decision in the later section on relevance (5.1 below). Again, the same proposition can occur in more than one box, as would the expression asserting that an elephant was larger than an atom, for it is about both atoms and elephants, and should appear under both topics..

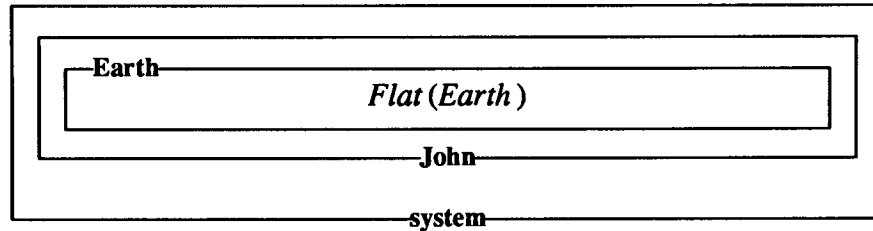
If the "topic" of a topic environment is a person (someone capable of having beliefs themselves), then the topic environment may contain, in addition to the beliefs about the person, a viewpoint environment containing particular beliefs held by that person about various topics. Normally and for obvious reasons of efficiency, this is only done for those beliefs of a given person that are, as some would put it, reportable, where that will often mean beliefs that conflict with those of the system itself. For example, suppose the system had beliefs about a person called John who believes that the Earth is flat. This would be pictorially represented as follows:



The John viewpoint, shown as the box with "John" on the lower edge, is a *nested viewpoint*, as it is enclosed within the system viewpoint shown (through an intervening topic environment about John, shown as the box with "John" on its upper edge). For simplicity, in the diagram of a nested viewpoint we often leave out propositions that are not in the innermost topic box: in the above example we would leave out the beliefs that John is a man, and that he is six feet tall. Further simplifying this, we often leave out all but the innermost topic box, leaving only it and the viewpoint boxes.

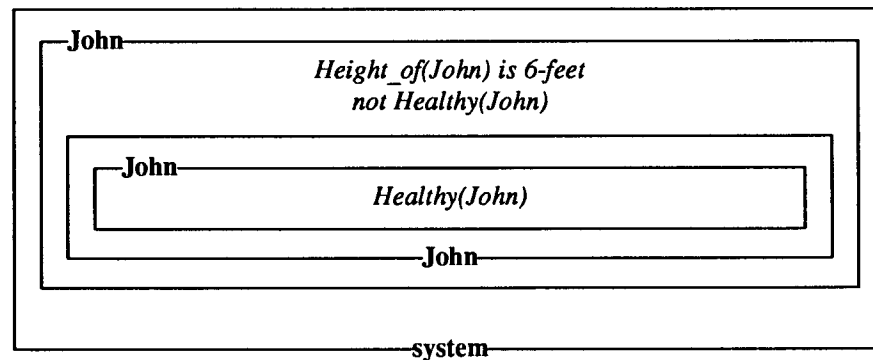


Hence, the above diagram would be simplified as:

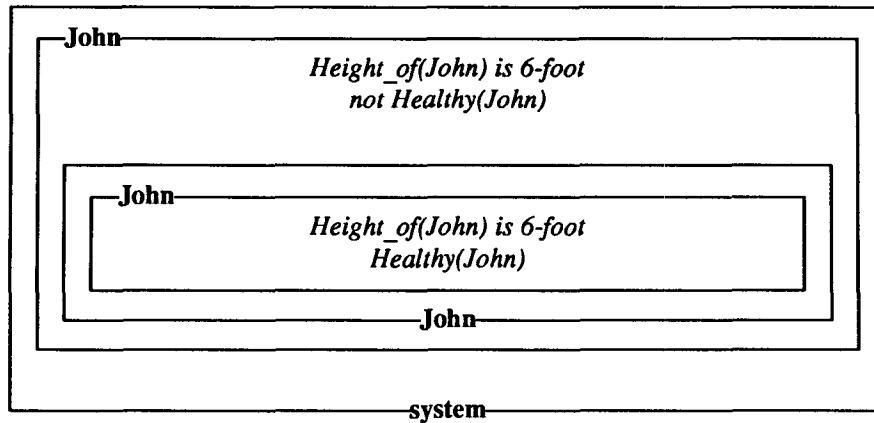


The system stores its own beliefs, and the beliefs of other agents that differ from the system's own beliefs. Others' viewpoints are generated on demand, a position we find both computationally and psychologically more plausible than the "prestored nesting" view mentioned above (Section 1). The process of generating a viewpoint can be regarded as an *amalgamation* mechanism that ascribes beliefs from one viewpoint to another (or, "pushing one environment down into another"); ascribing certain beliefs, transforming some, and blocking the ascription of others.

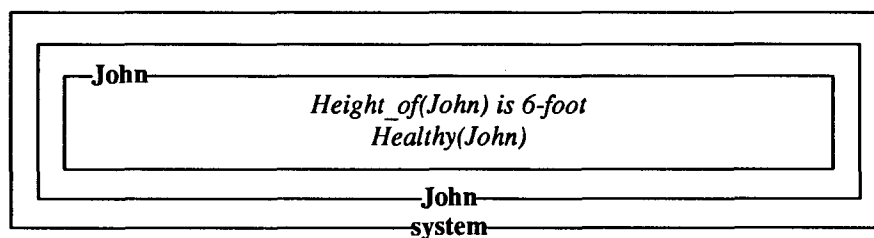
The simplest form of this algorithm, described in Wilks & Bien (1979, 1983), is that a viewpoint should be generated using a default rule for ascription of beliefs. The default ascriptional rule is to assume that another person's view is the same as one's own *except where there is explicit evidence to the contrary*. An important special case of such examples is when the topic is the same as the agent, and we can illustrate with that. Suppose that at a certain stage in dialogue the system, acting as a medical diagnostician, has the view of John that he is not healthy, and is six feet tall, while he believes himself to be healthy. We shall delay until the section on intensional objects below any question as to whether the system's John and John's own could be different entities. This basic situation is represented pictorially as follows:



The more complete environment for the system's view of John's view of himself can be generated by trying to ascribe the beliefs from the system's topic environment about John to the topic environment about John within John's viewpoint (where, as always, the last expression must be glossed as "the system's view of.."). One of the two beliefs survives the attempt but the other is blocked, giving the following state:



This can be pictured in the simplified (or as we shall call it, *compressed*) manner as



We see that in examples of this sort, where the topic is also the agent into whose environment an ascription is being attempted, propositions in an outer topic environment E are *pushed inwards* into a topic environment (for the same topic) within a viewpoint nested within E. Such inward pushing is central to our later observations of intensional identification and metaphor.

The above example demonstrates the basic ascription algorithm and a simple case of ascriptions being blocked. However, belief ascription is a far more complex phenomenon and the key to our method is the delimitation and treatment of cases where the default algorithm is incorrect. But even the default algorithm itself requires for its operation a notion of blocking beyond that of explicit contradiction: for example, the proposition *Healthy(John)* should be able to block *Sick(John)*, if *Sick* and *Healthy* are known to be incompatible predicates. Similarly, we appeal below to blocking that arises from incompatible function values, as in the blocking of “*Eye-colour(Frank) = Green*” by “*Eye-colour(Frank) = Blue*”. The more significant complication is that there is an entire class of beliefs that require the opposite of the default ascription rule given above. We call these *atypical beliefs* and they include technical expertise, self-knowledge (itself a form of expertise), and secrets. For example, beliefs that I have about myself, such as how many fillings I have in my teeth, are beliefs that I would not normally ascribe to someone else *unless I had reason to do so* (if, say, the person, to whom I was ascribing the belief, was my dentist). A representation based on lambda expressions is used in dealing with atypical beliefs, and is described elsewhere (Ballim, 1987; Ballim & Wilks, forthcoming; Wilks & Ballim, 1987), and follows a suggestion originally made by McCarthy and Hayes (1969). This combination of a basic default ascription rule, augmented by a mechanism for dealing with atypical belief, is an original algorithm and has not, to our knowledge, been described or tested elsewhere in the literature.

The essential feature of this notation is that lambda-expressions, as in the following example

*Cure-for(tuberculosis) is ((λx(Cure-for x)) tuberculosis)*

can only be evaluated by qualified believers (e.g., physicians in this case) in appropriate environments. Yet anyone can believe the Fregean triviality that the above sentence expresses unevaluated (and it is vital that they can) but a non-

trivial interpretation can only be placed on it by those who can evaluate the lambda expression in an environment. In a crude sense therefore, the lambda-expressions correspond to intensional representations and their evaluations, when available, to extensions, or at least other intensions in those situations where the evaluation of such an expression produces yet another lambda expression (see also Maida 1983).

The above expression, for example, might evaluate to another lambda expression using a predicate *Sulfonamide-drug*, for whose evaluation a particular drug might be an appropriate. Each evaluation would require an environment whose "holder" was qualified to perform it. It is really this possibility of successive evaluations of expressions that justifies the abstraction capacity of the lambda notation, since it could well result in expressions, such as a conjunction of predicates, for which there is no single predicate name. In conclusion, it is this general mechanism in *Viewgen* that deals with the problem of the over-application of the main default rule of ascription, since the ascription of unevaluable expressions, about, say, the number of my own teeth to you, does not lead to undesirable results.

### Intensional Objects and Their Identification

It is natural in a system of partitioned environment notation to treat environment boxes as intensional objects: to treat the Jim-object, pushed down into the Frank-object, as not just yielding by computation a set of beliefs that is Frank's-view-of-Jim, but also as a sort of intensional object we might call Jim-for-Frank. Let us now consider two simple cases of intensional objects and see how the basic default algorithm deals with them:

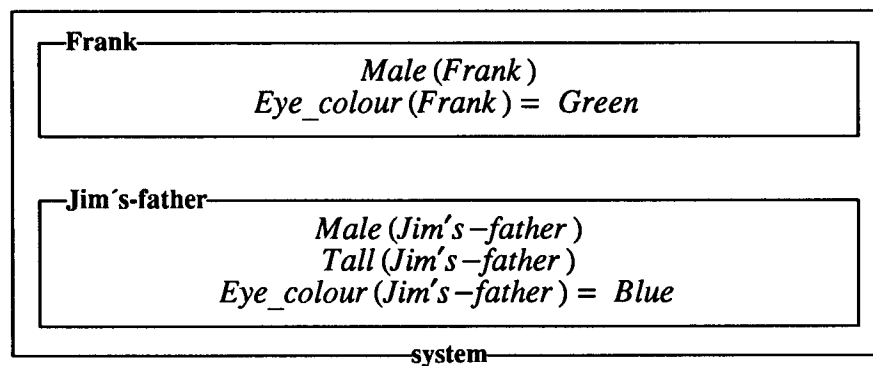
**CASE 1 (or Two-for-me-one-for-you):** The system believes that Frank and Jim's-father are two people, but that Mary, whose point of view is being computed, believes them to be the same person.

**CASE 2 (or One-for-me-two-for-you):** Mary believes Frank and Jim's-father to be separate people, whereas the system believes them to be the same individual.

Scenarios such as these are common, and arise over such mundane matters as believing or not believing that Johns-house is the same as the house-on-the-corner-of-X-and-Y-streets.

#### Two-for-me-one-for-you

Processing of first case will begin with the system having three topic environments: for Frank, Jim's-father and Mary. Two questions that arise are: What intensional object(s) should Mary's viewpoint contain? And what should be the beliefs about those intensional objects? Let us say that the system has beliefs about Frank and Jim's father as shown below.



The first question can be rephrased as "given certain intensional objects in one viewpoint (the system, in this case), what are the corresponding intensional objects in the system's version of another viewpoint (Mary's)?" Extending the normal default rule for belief ascription to cope with intensional object ascription, we would say, naturally enough, that intensional objects in one environment directly correspond to intensional objects in another environment, *unless there is counter evidence to believing this*. This notion of correspondence of intensional objects between environments can be expressed as beliefs, but these beliefs must be of a type different from those we have previously discussed.

Although it is true that beliefs about intensional correspondence *are* beliefs about someone, for instance that the system believes about Mary that her intensional objects correspond to its own in a particular way, they clearly cannot be

beliefs of Mary herself. Referring to our particular example, we say that the system's belief that Mary has a single intensional object corresponding to the system's Frank and Jim's-father objects is a belief *for* Mary.

These beliefs of intensional correspondence are beliefs of one agent that describe how the agent's intensional objects correspond to intensional objects *for* another agent. Thus, they shall be referred to as *beliefs for* an agent P, and we shall place them in a separate partition as diagrams below will illustrate. A reader might feel here that there is no need for an additional box type, but further reflection may correct this. This relationship cannot be captured by co-reference (as opposed to what we are calling correspondence) since within these intensional contexts entities are not coreferential, *ex hypothesi*. The For-box (see Fig. below, where it should be noted that the ordered arguments that correspond for-Mary are precisely not Konolige's "views", even though the notation is the same.

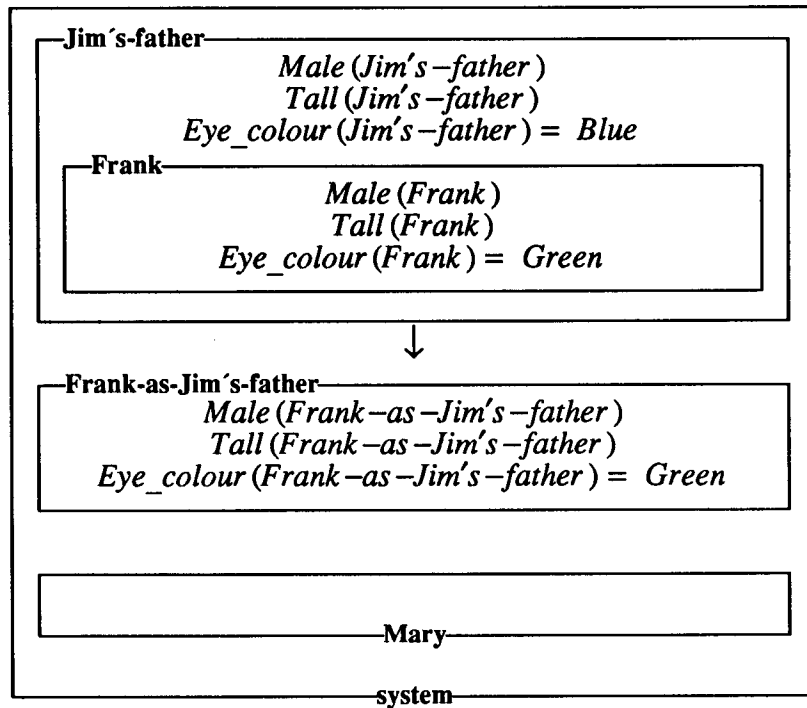
In the case at hand (CASE 1), Mary's viewpoint ends up containing a single intensional object O (a topic environment) corresponding both to the system's Frank object (topic environment) and to the system's Jim's-father object (topic environment). The question now is to decide what should be put inside the environment O. One possibility is to combine the information in the system's Frank and Jim's-father objects *symmetrically*, removing any conflicting information. In the present case, this would result in O stating that Frank/Jim's-father is male and tall, but stating neither that he has blue eyes nor that he has green eyes. However, we claim that *in realistic situations it will often be more appropriate to take an asymmetrical view*, in which we choose to give precedence either (i) to the information in the system's Frank object over the information in the system's Jim's-father object, or (ii) vice versa. Choice (i) reflects the presumption that there is a stronger or closer correspondence between Mary's idea of Frank and the system's idea of Frank than there is between her idea of Frank and the system's idea of Jim's father. This difference of closeness would be plausible, for instance, if the system regarded Mary's view of Frank as being essentially the same as its own except in making the (presumed) mistake of taking Frank to have the property of being Jim's father.

Choice (ii) reflects the converse presumption, which would be most likely to arise from a hypothesis that Mary is focussing on the person-description "father of Jim", and that she happens to hold that this description identifies Frank. Our claim is that in realistic situations there is more likely to be a reason for making one of these choices than to take the symmetrical approach.

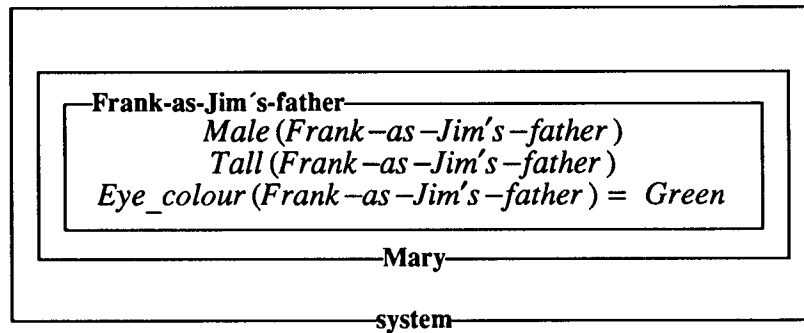
We handle the asymmetrical choices as follows. For choice (i), the system constructs an intensional object O called "Frank-as-Jim's-father" inside Mary's viewpoint. This object is so-called because it is, so to speak, "the Jim's-father view of Frank" (according to Mary). Notice that we are *not* here saying that the object is the view of Frank that Jim's father *holds* (according to Mary); rather, the object is a view of Frank that is coloured by the idea that he is Jim's father. This way of regarding Mary's intensional object O is directly reflected in the proposed process for constructing O, as we shall see in a moment. Mary's Frank-as-Jim's-father object, O, arises in two stages, as follows.

*Stage 1:* The system's view of Frank as Jim's father is created. This view is created as a topic environment O' inside the system's viewpoint. The creation occurs in three substages:

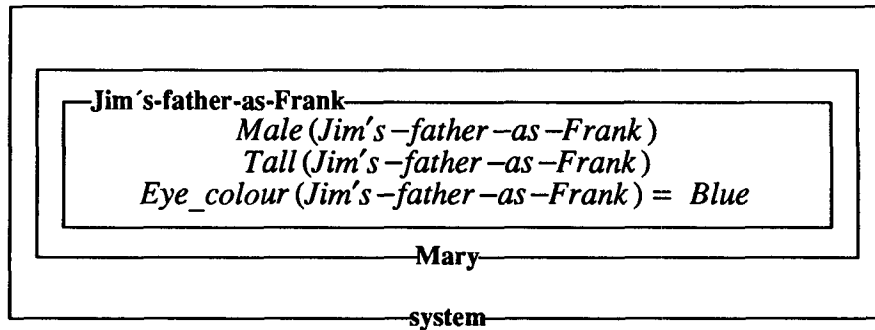
- (1a) Initially, a copy of the system's Frank object (topic environment) is placed inside the Jim's-father object (topic environment), as shown in the next figure. Intuitively, the idea so far is that we have not yet tried to identify Frank as Jim's father, but have merely established a view of Frank that is, so to speak, in the context of Jim's father. That context does not have an effect until substage (1b).
- (1b) We now respect the required identification of Frank as Jim's father. We try to push the beliefs in the system's Jim's-father object into the Frank object embedded within it, *using the ordinary default rule*, with the slight modification that *Jim's-father* is replaced by *Frank* in a pushed belief. Thus, the beliefs that Jim's father is male and is tall are successfully pushed in (although the former happens to duplicate a belief already in the embedded Frank object), but the belief that Jim's father has green eyes is blocked by the blue-eye belief already in the embedded Frank object.
- (1c) The final substage in constructing the system's Frank-as-Jim's-father object O' is to pull out the Frank object that is embedded within the Jim's-father object, making it into an object (topic environment) O' at top level within the system's viewpoint. In doing this we replace the "Frank" topic-name by the name "Frank-as-Jim's-father", and similarly change the *Frank* symbols inside the environment to *Frank-as-Jim's-father*. The following diagram shows the result, with the arrow notation indicating the pull-out process.



Stage 2: We now ascribe the system's beliefs about Frank as Jim's father — that is, the beliefs inside  $O'$  — to Mary, *once again using the ordinary default rule*. On the assumption that there is no prior information about Mary's view of Frank/Jim's-father (e.g. that his eyes are brown), all that will happen is that a copy  $O$  of  $O'$  will be created inside the Mary viewpoint, giving the revised Mary-viewpoint shown in the following figure.



If we had had prior information from discourse input that Mary believes the person's eyes to be brown, then there would already have been a Frank-as-Jim's-father object (topic environment)  $O$  inside Mary's viewpoint, and the beliefs in  $O'$  would all have got pushed into that object except for the green-eye belief. If the system had decided to give precedence to the Jim's father information rather than to the Frank information in doing the intensional identification (that is, if it had made choice (ii) above) then it would have generated the following state by an analogous process:

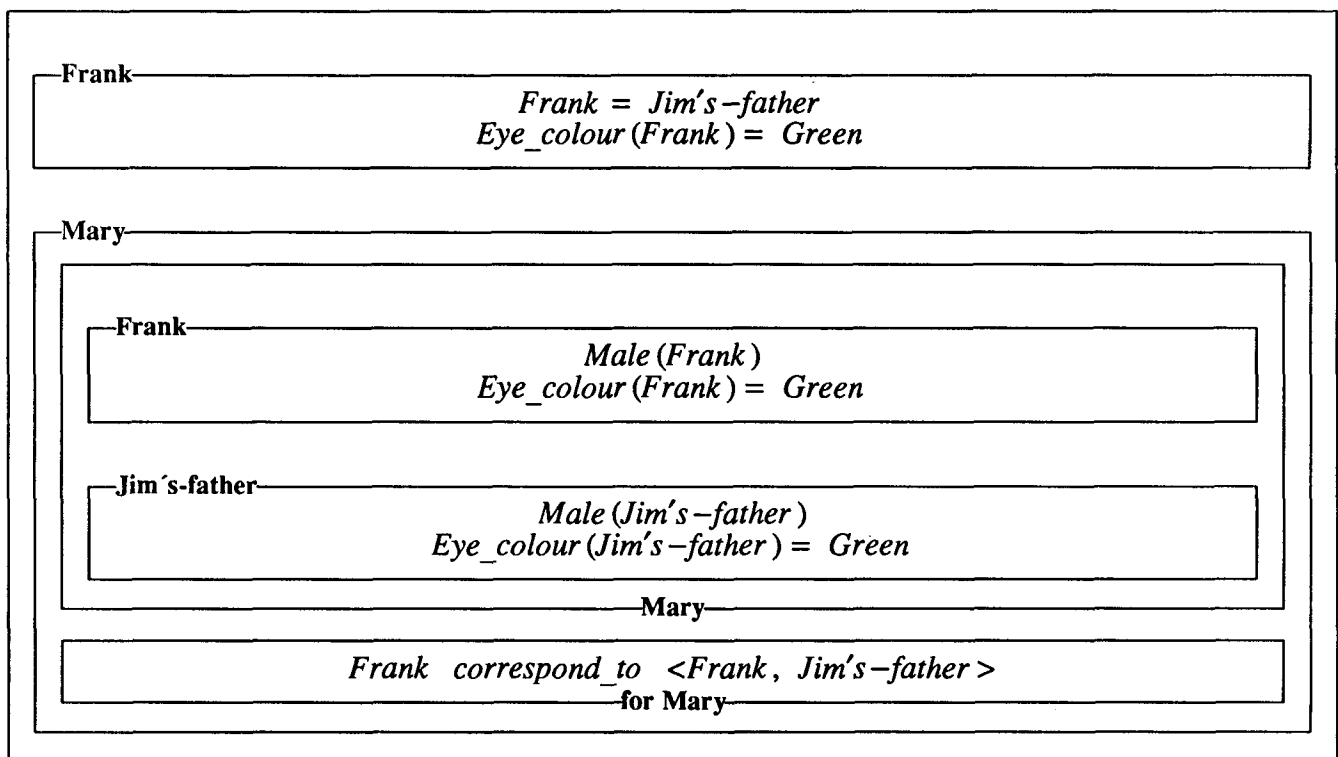


It might be thought that a symmetric intensional object, with the feature differences appearing as disjunctions (e.g. Eye\_color Blue OR Green) would be appropriate as a construct for the Mary environment. We suggest that this is in fact psychologically less plausible, and that subjects do construct stronger, and more refutable, hypotheses. A final important thing to notice about the process described above is that the crucial pushing of information from the Jim's-father environment into the embedded Frank environment (or vice versa) is exactly the type of "inward" pushing used in the basic examples with which we illustrated basic belief ascription in section 2.

In Sections 6 and 8 we shall seek to show that belief ascription (e.g. Jim's-father's-view-of-Frank), intensional identification (e.g. Frank-as-Jim's-father), and even metaphor are all different forms of a single fundamental computational process.

**Treatment of CASE 2: One-for-me-two-for-you**

In the second case, where the system believes in one individual but Mary two, then the natural computation of Mary's view of either Frank or Jim's-father is simply to push the system's single representation, changing "Frank" to "Jim's-father" as necessary. This is shown in the following figure.



These are not merely aliases, but are best thought of as dual ascriptions, performed by making two identical copies. Further information about Mary's beliefs would then presumably cause the contents of the two environments to differ, since she presumably has at least some differing beliefs about what she believes to be distinct individuals.

### Discussion

Neither CASE 1 nor CASE 2 turns out to be particularly problematic, and the situation is no different if the entities about whose identity there is dispute are non-believers rather than believers. Those would be like the classic but natural cases such as a difference between dialogue participants as to whether Tegucigalpa and Capital-of-Honduras are, or are not, the same; or as to whether Rome or Avignon should be identified with City-of-the-Popes.

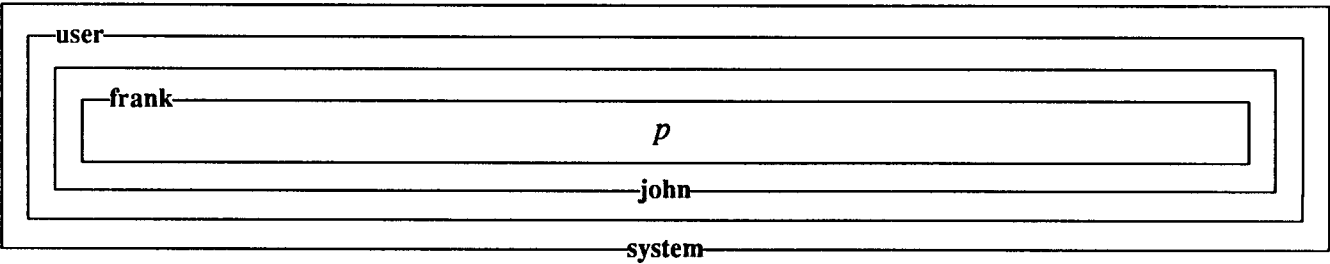
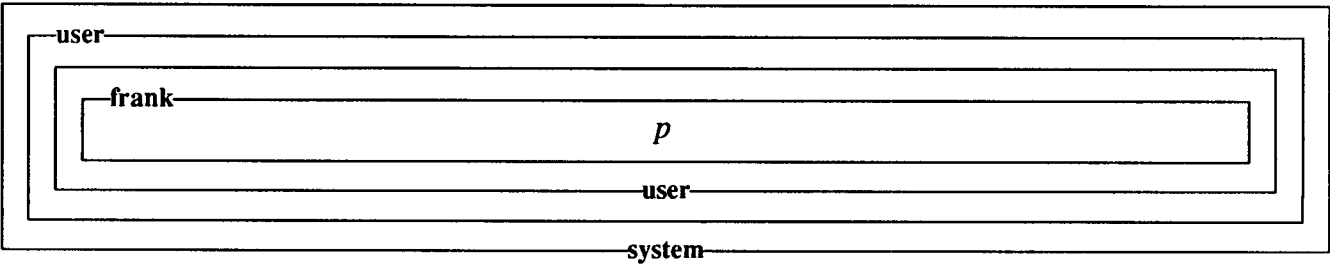
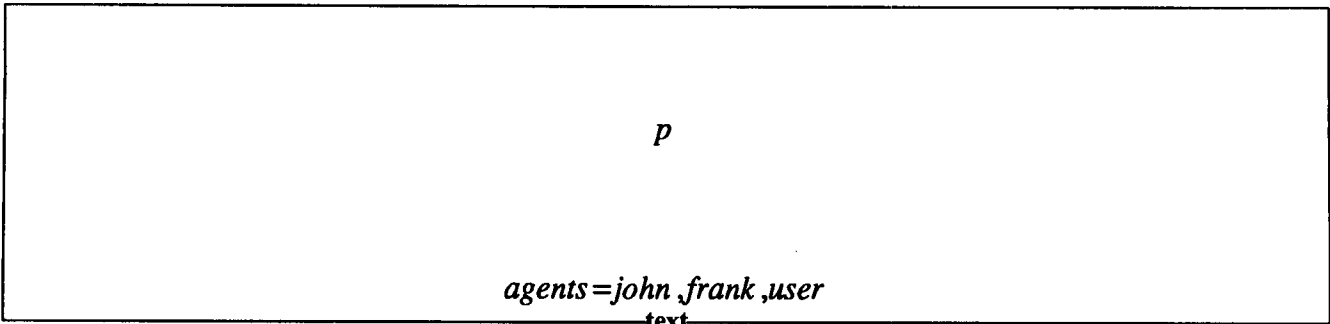
More difficult cases, that bring in all the panoply of philosophical distinction and discussion are those conventionally discussed under the *de re/de dicto* distinction. One type is the following: the system reasonably believes Feynman to be a famous physicist but encounters Frank who, on the strength of a single appearance on the TV screen, believes him to be a famous TV performer. For the sake of this example, it is essential to accept that the two occupations are incompatible. Suppose the discussion now forces the system to construct its view of Frank's view of Feynman. Now, there will be no point at all in performing that computation unless the system believes Frank's beliefs to be *de re*. Frank no doubt considers his own beliefs *de re*, as we all do. The crucial thing is that the system believe this, and the test would be some proposition in the Frank environment, and ABOUT Frank, equivalent to ("*Feynman*" *names Feynman*). If that is not present, the system should infer that Frank has another person in mind: that his beliefs are *de dicto* FOR THE SYSTEM, and hence any pushdown computation would be pointless.

Consider the relation of this example to the former, simpler, cases, where the system can identify or separate distinct environments. This last case would be like that if the system knew which non-Feynman individual Frank was confusing Feynman with, perhaps Johnny Carson. In that case, the system could perform a pushdown, even though it believed Frank's beliefs to be *de dicto* as far as Feynman was concerned, for they would be *de re* with respect to Johnny Carson. The system could then push Carson into Frank, while changing the resulting environments name to "Feynman". To summarize, the absence of ("*Feynman*" *names Feynman*) in the Frank environment is only a reason for not pushing down Feynman, but leaves open the possibility of some other *de re* push down.

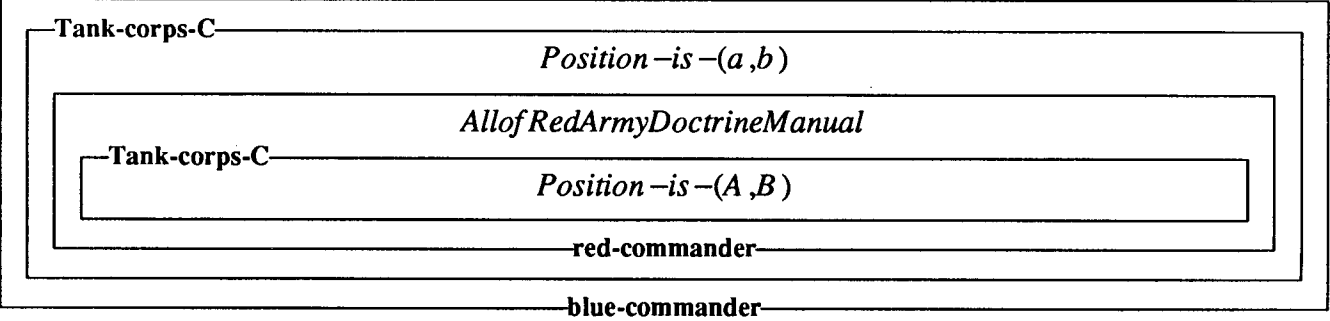
## UNIFYING VIEWGEN AND MGR

Both ViewGen and MGR are systems producing multiple models or views of a situation, but only in the ViewGen system does that multiplicity depend on separate environments representing the views of others explicitly: i.e. the multiplicity of ViewGen's models are to be mapped to different agents, while in MGR the multiplicity of models are the alternatives present in a single agent. Hence the multiplicities of the two theories nest in a natural way, and, we hope to show (although our project is still in its early days) combine to filter each other's products, rather than multiply them.

In its earliest incarnation (Wilks & Bien 1983) ViewGen was also seen as a provider of multiple alternative models of text or message content. So, in the following diagram, adapted from that paper, we can consider a message *p* in a text (first frame of diagram) that mentions the agents user, john and frank. The analysis system can then be considered (by processes not set out in detail here) as creating the two nested environments shown in the two succeeding frames, each of which contains a representation of *p*, plus whatever beliefs and goals are appropriate to that inner environment. hence, given that those two sets differ significantly, the consequences, in those two environments, of *p* and whatever other beliefs are in there, will, taken together, be significantly different, and will be effectively different interpretations of the original message *p*. This ambiguity is not syntactic or semantic ambiguity but pragmatic ambiguity against differing assumptions.

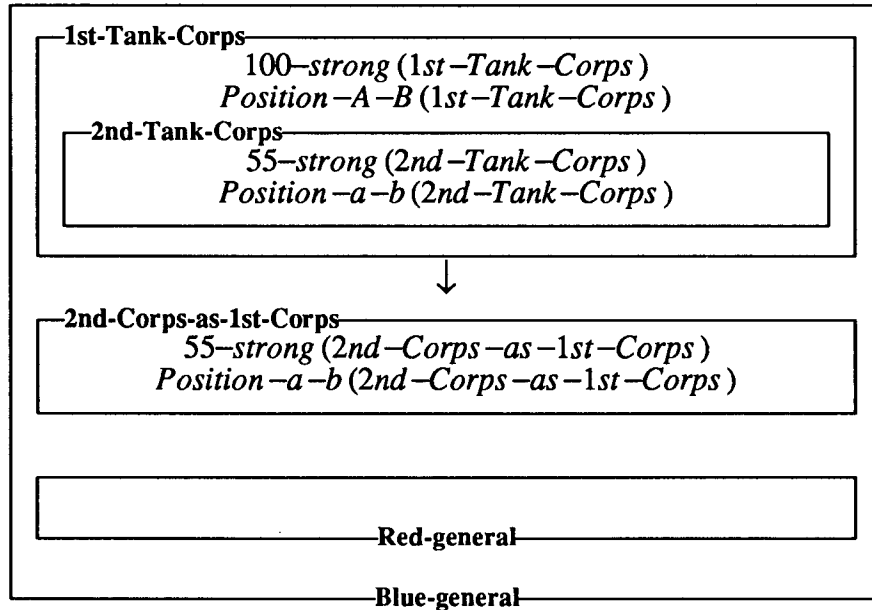


The natural application of that in, say, an Army battle scenario where message-derived intelligence reports and stored doctrine provide the data for model construction is one where the Blue has reason to believe the Red commander is crucially wrong about, say, the position of 22nd Tank Division, a belief that may be accidental or by Blue design and is derived from a parsing of Red messages:





The ability of the Blue Commander to compute effectively in this scenario also depends on Blue's access to Red Army doctrine manuals, which we have to imagine having been reduced to machine-readable form in some way. An important variant on this theme, for which ViewGen provides the representational mechanism, is a situation where, in Blue's view, the new commander has confused and conflated two Blue Tank Divisions, say 1st and 2nd. The problem then, as we noted in the ViewGen section above when discussing Jim-as-Frank's-father is how to have a representational entity for an object that Blue, in this case, does not believe in the existence of, namely the wrongly conflated 1st and 2nd Tank Divisions. It will be remembered that it is crucial, in our view that this be an asymmetric representation, not some simple union of properties.



Our principal architectural aim is to combine these techniques into a system that can play a role within the standard Army intelligence analysis and battle management infrastructure that can be diagrammatically represented as follows:

DIAGRAM: DATA AND GUIDANCE

The G2 is an officer whose job is to evaluate various types of field data (i.e. weather maps, field observations, sensor outputs, etc.) and utilizing the 'Red Doctrine' as a set of guidelines, provide explanations or plausible situations to the

commander. In other words, his job is to integrate the field data with the Red Doctrine and inform the commander of all possible explanations of the data. These explanations should be well ordered (i.e. weighted), in order to better inform the commander.

The Red Doctrine is a synthesis of all experiences or facts regarding the operations of the Red Army. Theoretically the Red Doctrine is written to anticipate all possible situations, including elements of deception and provide appropriate responses for them. We see our project as developing an AI system to act as another computer-based tool for the G2 to use. The primary system under development, the Air-Land Battle management (ALBM) system has proven to be strong when it comes to handling troop movements in different terrain and in planning a response, but weak in hypothesis generation of enemy intentions.

The goal of this project is for MGR to produce the range of all possible situations (called models in the case of MGR) based on the data and the Red Doctrine. The models will be hypothetical in nature because the input data are noisy, thus requiring interpretation to explain them. MGR looks for coherent subsets of data to produce interpretations. The more uncertain the data the greater the number of models, and these models may have components that are incoherent with other models. In general multiple models may originate from any of three possible sources:

The data is incoherent.

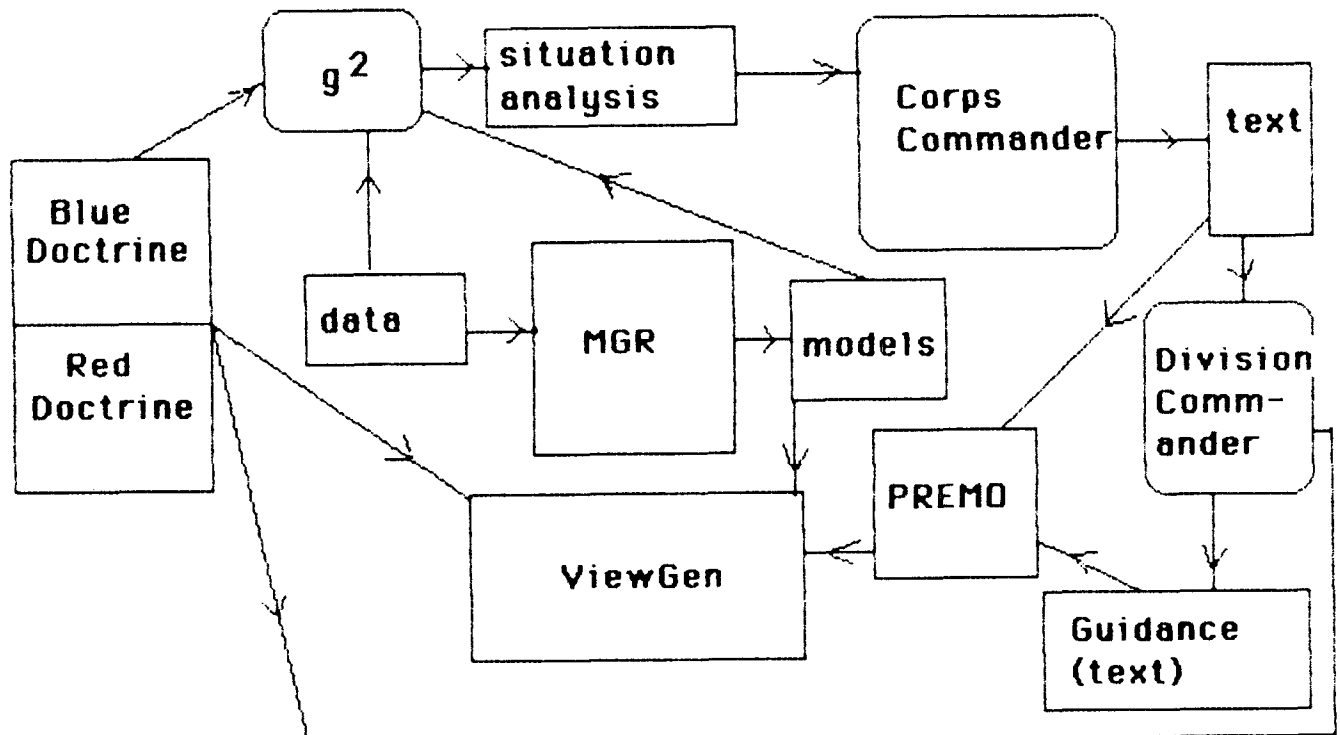
The Doctrine is too general, thus allowing coherent alternative interpretations.

MGR can generate alternatives from the ambiguity inherent in existentially quantified descriptions.

The assembled information produced by a G2 and the models produced by MGR should be consistent for the same data. Each model or set of alternate models can be used as the basis of prediction of future enemy movements, tactics etc. The use of models for both explanations and prediction is an essential element in MGR's use. Let us finally, and tentatively, turn to possible ways of integrating these techniques into a more complex system of command and advice.

We envisage a system in which the PREMO parser takes textual (e.g. Red or Blue Army doctrine) or message input and produces an output in Conceptual Graph format. This output is fed to MGR which generates multiple models to explain or represent the input. The output of MGR is then represented *when required but not in general* as different nested environments in ViewGen and then evaluated to determine which surviving environment best fits the data (i.e. is most coherent) or what to do if things go wrong.

A suggestion for integrating ViewGen and the PREMO parser into the existing MGR project is shown below.



In the diagram, there are several believers at work interpreting data and producing responses. Roughly speaking, and without any automated help, the sequence is as follows: the Corps G2 assembles all available data and produces a set of (weighted) interpretations as to Red's intentions. These are produced from the incoming field data and from his knowledge of Red doctrine, which stipulates default tactics and maneuvers in a wide range of different situations. The G2's reports are received by the Corps commander and he chooses the basis for a Blue plan based on the situation analysis and his knowledge of Red doctrine, and, for planning purposes, Blue doctrine. The commander issues "commander's guidance" reports which are then passed on to the division commanders as orders. The division commanders then re-interpret these guidances in order to make their plans, in the light of their knowledge of both Red and Blue doctrine.

There are several belief systems at work, and plenty of room for inconsistency to occur. The clearest area for this to arise is not in the different interpretations of Red doctrine, but in interpreting the input. In the case of the G2 this is interpreting the data. For the Corps commander, this is interpreting the G2's reports, and for the division commanders this is interpreting the Commander's guidance. These different interpretations stem from different applications of the same knowledge to the input data. The source of the differences could be postulated to be due to various psychological states (mistrust, pig-headedness etc.) but even in the "objective" case different weighing of evidence according to personal experience is bound to occur.

MGR is "objective" since it has no possible psychological leanings, but it still only produces one possible set of views of the battle. The hope is to make MGR produce more objective alternatives than analysts commonly do, but its inclusion in this chain of command is only one factor in the plans actually produced and carried out by the army. However, its strength is that it is, at base, algorithmic, and therefore its output is reproducible. It can be therefore be used as a benchmark against which other agents' beliefs can be measured. One way to do this is to nest MGR's "beliefs" at the core of a nested sequence of belief spaces. Successively these would be G2, Corps Commander, Division Commander. These spaces would be maintained by ViewGen by importing the relevant changing data at each level, and recomputing the set of beliefs accordingly. The nesting of belief spaces then allows for consistency checking between the agents, using MGR's output as a base case in order to evaluate the inconsistencies as they arise.

As an example, consider a piece of intelligence that places a Red MRD (Motorized Rifle Division) close to a known avenue of approach to a potential battle front. MGR would say (objectively) that the division would end up at the front on that avenue simply based on current activity (direction and speed of movement) and the intervening terrain. It would also hypothesize that this division could either play a main or support role in the attack, in conjunction with the other divisions known to be in the area. The G2's job is essentially to produce such impartial explanations, albeit weighted by likelihood. Now postulate that the Corps commander has detected (perhaps much earlier in the battle) that this particular division has been behaving strangely (he may, for instance know the personal preferences of the division commander - something that MGR would not know). He may therefore choose to ignore the G2's analysis and assume, for instance that the division will play no major part in the battle - i.e. for planning purposes it does not exist. His guidance will then reflect this interpretation, providing one level of inconsistency between this guidance and MGR's (or the G2's) output. Now assume that one of Blue's division commanders is a little "trigger happy" and is itching for a skirmish. He could detect the Corps commander's discounting of the strangely behaving Red division, and choose to bend the orders sufficiently to force an encounter with it. His orders will then reflect another level of inconsistency, this time with the Corps commander's guidance. He may, for instance, believe that the rogue division is an easy target, even though his orders are to ignore it.

ViewGen can handle this multiply nested set of inconsistencies, and with MGR output as the impartial 'base base' can give judgements as to the direction of the inconsistencies, without prejudging the rights or wrongs of the case. Without MGR the only judgements that can be made are relative, and might be of little use in a real case. Several questions come to mind as to how to proceed. Firstly, whereas MGR produces interpretations according to a default set of ideas about Red doctrine and how to handle raw data, the belief spaces of individuals is much harder to capture and to model. ViewGen cannot create beliefs, only modify statements according to a system of belief already captured and represented. Secondly, The text messages that are handed down the chain of command may not turn out to adequately reflect the influence of the agent's belief space, and that another round of interpretation may be called for before they can be used by ViewGen. Our preliminary examination of sample Commander's guidance messages is inconclusive in this regard. Thirdly, although Red doctrine is reasonably easy to represent, beliefs about how it should be used may not be. MGR presents a fairly neutral view of this, preferring to err on the side of producing too many models that can be filtered out later, rather than missing a potentially important alternative. These differences in inference strategy can be incorporated into ViewGen's evaluation mechanisms, but exactly how is an issue for further research.

## References

- Allemang, D., Tanner, M., Bylander, T., Josephson, J. (1987) On the computational complexity of hypothesis assembly. In *Proceedings of IJCAI87*, Milan, Italy.
- Ballim, A. (1986) Generating point of view. *Memoranda in Computer and Cognitive Science*, MCCC-86-68. Computing Research Laboratory, New Mexico State University, Las Cruces, New Mexico 88003, USA.
- Ballim, A. (1987) Subjective ascription of belief to agents. In *Advances in artificial intelligence*. Edited by J. Hallam & C. Mellish. Chichester, England: John Wiley & Sons.
- Ballim, A. (1988) A language for representing and reasoning with nested belief. *First Annual Irish Conference on Artificial Intelligence and Cognitive Science*. Dublin, Ireland.
- Ballim, A. and Wilks, Y. (forthcoming) *Artificial believers*. Hillsdale, N.J.; Lawrence Erlbaum Associates.
- Barnden, J.A. (1983) Intentions as such: an outlined. *Proceedings of the 8th International Joint Conference on Artificial Intelligence*. Los Altos, Calif: Morgan Kaufmann.
- Barnden, J.A. (1986a) A view point distinction in the representation of propositional attitudes. *Proceedings of the 5th Nat. Conf. on Artificial Intelligence (AAAI-86)*. Philadelphia: Morgan Kaufmann.
- Barnden, J.A. (1986b) Imputations and explications: representational problems in treatments of propositional attitudes. *Cognitive Science*, 10, (3), pp.319-364.
- Barnden, J.A. (1987a) Interpreting propositional attitude reports: towards greater freedom and control. In *Advances in artificial intelligence - II*. Edited by B. du Boulay, D. Hogg & L. Steels. Amsterdam: Elsevier (North-Holland).
- Barnden, J.A. (1988a) Propositional attitudes, commonsense reasoning, and metaphor. *Proceedings of the 10th Annual Conference of the Cognitive Science Society*. Hillsdale, N.J.: Lawrence Erlbaum Associates.
- Barnden, J.A. (1988b) Propositional attitudes, polysemy and metaphor: initial report. *Memoranda in Computer and Cognitive Science*, MCCC-88-139. Computing Research Laboratory, New Mexico State University.
- Boden, M. (1977) *Artificial Intelligence and Natural Man*. Hassocks, UK: Harvester Press.
- Boguraev, B. K. (1979) Automatic Resolution of Linguistic Ambiguities. Technical Report No.11, University of Cambridge Computer Laboratory, Cambridge, England.
- Carter, D. (1984) An Approach to General Machine Translation Based on Preference Semantics and Local Focussing. In *Proceedings of the 6th European Conference on AI (ECAI-84)*, pp.231-238, Pisa, Italy.
- Carter, D. (1987) *Interpreting Anaphors in Natural Language Texts*, Ellis Horwood: Chichester, England.
- Charniak, E. (1978) On the use of knowledge in language comprehension. *Artificial Intelligence*, 11, pp.225-265.
- Charniak, E. (1986) Motivation analysis, abductive unification and non-monotonic equality. *Artificial Intelligence*, 34: 275-295.
- Cohen, P. and Levesque, H. (1985) Speech acts and rationality. *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*. University of Chicago.
- Coombs, M. J. and R. T. Hartley (1987) The MGR algorithm and its application to the generation of explanations for novel events. *International Journal of Man-Machine Studies* 27: 679-708.
- Coombs, M. J. and R. T. Hartley (1988a) Explaining novel events in process control through model generative reasoning. *International Journal of Expert Systems* 1: 89-109.
- Coombs, M. J. and R. T. Hartley (1988b). Design of a software environment for tactical situation development. *Proceedings of the US Army Symposium on Artificial Intelligence Research for Exploitation of the Battlefield Environment*, El Paso, November 1988.
- Dinsmore, J. (1987) Mental Spaces from a functional perspective. *Cognitive Science*, pp.1-23.
- Fass, D. (1986) Collative Semantics: An Approach to Coherence. Memorandum in Computer and Cognitive Science, MCCC-86-56, Computing Research Laboratory, New Mexico State University, New Mexico.
- Fass, D. (1987) Semantic Relations, Metonymy, and Lexical Ambiguity Resolution: A Coherence-Based Account. In *Proceedings of the 9th Annual Cognitive Science Society Conference*, University of Washington, Seattle, Washington, pp.575-586.

- Fass, D. (1988a) Collative Semantics: A Semantics for Natural Language Processing. Memorandum in Computer and Cognitive Science, M CCS-88-118, Computing Research Laboratory, New Mexico State University, New Mexico.
- Fass, D. (1988b) Metonymy and Metaphor: What's the Difference? *Proceedings of the 12th International Conference on Computational Linguistics (COLING-88)*, Budapest, Hungary.
- Fass, D. (1988c) An Account of Coherence, Semantic Relations, Metonymy, and Lexical Ambiguity Resolution. In Steve L. Small, Gary W. Cottrell, and Michael K. Tanenhaus (Eds.) *Lexical Ambiguity Resolution in the Comprehension of Human Language*, Morgan Kaufmann: Los Altos, CA, pp.151-178.
- Fauconnier, G. (1985) *Mental Spaces: aspects of meaning construction in natural language*. Cambridge, Mass.: MIT Press.
- Fields, C. A., M. J. Coombs and R. T. Hartley (1988a). MGR: An architecture for problem solving in unstructured task environments. *Proceedings of the Third International Symposium on Methodologies for Intelligent Systems*, Elsevier, Amsterdam, 44-49.
- Fields, C. A., M. J. Coombs, E. S. Dietrich, and R. T. Hartley (1988b) Incorporating dynamic control into the Model Generative Reasoning system. *Proc. ECAI-88*, 439-441.
- Gershman, A. (1977) Knowledge Based Parsing. Memorandum from Yale University Computer Science Department.
- Hobbs, J.R., Stickel, M., Martin, P., Edwards, D. (1989) Interpretation as Abduction (Draft). AI Center, SRI International.
- Huang, X. (1984) The Generation of Chinese Sentences from the Semantic Representations of English Sentences. In *Proceedings of International Conference on Machine Translation*. Cranfield, England.
- Huang, X. (1988) XTRA: The Design and Implementation of A Fully Automatic Machine Translation System. Memorandum in Computer and Cognitive Science, M CCS-88-121, Computing Research Laboratory, New Mexico State University, New Mexico.
- Josephson, J.R., B. Chandrasekaran, J. W. Smith, and M. C. Tanner (1987) A mechanism for forming composite explanatory hypotheses. *IEEE Trans. on Systems, Man and Cybernetics* 17: 445-454.
- Konolige, K. (1983) A deductive model of belief. *Proceedings of the 8th International Joint Conference on Artificial Intelligence*. Los Altos, Calif.: Morgan Kaufmann.
- Levesque, H. (1989) A knowledge-level account of abduction. In *Proceedings of IJCAI89*, Detroit.
- Maida, A.S. (1983) Knowing intensional individuals. *Proceedings of the 8th International Joint Conference on Artificial Intelligence*. Los Altos, Calif.: Morgan Kaufmann.
- Maida, A.S. (1984) Belief spaces: foundations of a computational theory of belief. *Tech.Rep. CS-84-22*. Dept. of Computer Science, The Pennsylvania State University.
- Maida, A.S. (1986) Introspection and reasoning about the beliefs of other agents. *Proceedings of the 8th Annual Conf. of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Maida, A.S. (1988) A syntactic approach to mental correspondence. *Proceedings of the Canadian Soc. for Computational Studies of Artificial Intelligence*. Edmonton, Alberta.
- Minsky, M. (1975) A framework for representing knowlege. In *The psychology of computer vision*. Edited by P.H. Winston. New York: McGraw-Hill.
- Nau, D. and J. Reggia (1986) Relationships between deductive and abductive inference in knowledge-based diagnostic problem solving, in L. Kerschberg (Ed.), *Expert Database Systems: Proceedings of the First International Workshop*. New York: Benjamin Cummings.
- Perrault, R. (1987) Invited presentation. *Meeting of the Association for Computational Linguistics* Stanford, Calif.
- Perrault, R. (forthcoming) An application of default logic to speech act theory. In *Plans and intentions in communication and discourse*. Edited by P. Cohen, J. Morgan & M. Pollack. Cambridge, Mass.: MIT Press.
- Perrault, R. and Allen, J. (1980) A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics*, 6, pp.167-182.
- Reiter, R., de Kleer, J. (1987) Foundations of assumption-based truth maintenance systems. In *proceedings of AAAI87*, Seattle.
- Shadbolt, N (1983) Processing reference. *Journal of Semantics*, 2, (1), pp. 63-98.

- Slator, B. (1988) Lexical Semantics and a Preference Semantics Parser. Memorandum in Computer and Cognitive Science, MCCS-88-116, Computing Research Laboratory, New Mexico State University, New Mexico.
- Slator, B. (1988) Constructing Contextually Organized Lexical Semantic Knowledge-Bases. *Proceedings of the Third Annual Rocky Mountain Conference on Artificial Intelligence*, Denver, CO, (June 13-15), pp.142-148.
- Slator, B. and Y. Wilks (1989) PREMO: parsing by conspicuous lexical consumption. In *Proceedings of the International Workshop on Parsing Technologies*, Carnegie Mellon U., Pittsburgh. Edited by M. Tomita.
- Wiebe, J. and Rapaport, W. (1986) Representing De Re and De Dicto belief reports in Discourse and Narrative. *Proceedings of IEEE*,74, pp. 1405-1413.
- Wilks, Y. (1972) *Grammar, Meaning, and the Machine Analysis of Language*, Routledge and Kegan Paul: London.
- Wilks, Y. (1975a) A Preferential Pattern-Seeking Semantics for Natural Language Inference. *Artificial Intelligence*, 6, pp.53-74.
- Wilks, Y. (1975b) An Intelligent Analyser and Understander for English. *Communications of the ACM*, 18, pp.264-274.
- Wilks, Y. (1978) Making Preferences More Active. *Artificial Intelligence*, 10, pp.75-97.
- Wilks, Y. and Ballim, A. (1987) Multiple agents and the heuristic ascription of belief. *Proceedings of the 10th International Joint Conference on Artificial Intelligence*. Los Altos, Calif.: Morgan Kaufmann.
- Wilks, Y. and Ballim, A. (1988) Shifting the belief engine into higher gear. In *Proceedings of the international conference on AI methodology and systems applications*. Edited by T.O'Shea & V. Sgurev. Amsterdam: Elsevier Publishers.
- Wilks, Y. and Ballim, A. (in press, a) Belief systems: ascribing belief. In *Proceedings of KIFS-87*. Edited by T. Christaller.
- Wilks, Y. and Ballim, A. (in press, b) The heuristic ascription of belief. In *Review of cognitive science, 1*. Edited by N.E. Sharkey. Norwood,NJ: Ablex.
- Wilks, Y. and Bien, J. (1979) Speech acts and multiple environments. *Proceeding of the 6th International Joint Conference on Artificial Intelligence*. Tokyo, Japan.
- Wilks, Y. and Bien, J. (1983) Beliefs, points of view and multiple environments. *Cognitive Science*,8, pp.120-146.
- Wilks, Y., Fass, D., Guo, C., McDonald, J., Plate, T., and Slator, B., (1988) Machine Tractable Dictionaries as Tools and Resources for Natural Language Processing. Edited by J. Pustejovsky, *Lexical and Computational Semantics*, Cambridge, MA: MIT Press.