A META-LEVEL APPROACH TO EXPLORATION OF MULTIPLE KNOWLEDGE BASES

Mandy Haggith

Department of Artificial Intelligence
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
Scotland, UK
Tel : (031) 650-2707
email : hag@aisb.ed.ac.uk

May 1994

For presentation at CKBS-94

This paper presents a meta-level framework for exploration of multiple knowledge bases in which conflicting opinions are expressed. In the first section, the usefulness of a meta-level approach to inconsistency handling is explained and comparisons made with other related work. In the second section a formal meta-level language is defined to allow arguments from knowledge bases to be reasoned about independently of their object-level representation. Four basic meta-level relations are introduced : *disagreement*, *equivalence*, *justification* and *elaboration*. In the third section, an example is given to illustrate the framework. The fourth section demonstrates how the four basic relations can be used to guide a user in exploring the range of opinion in a collection of knowledge bases. The notion of a *conflict set* is introduced to provide a focussed 'window' on the particular disagreements a user is interested in. A flavour is also given of how these primitive relations can be used as the basis of definitions of more complex argument constructs, such as those encountered in the literature on argumentation. These argument constructs may provide ways to automate comparison and evaluation of knowledge bases. An initial prototype has been implemented to evaluate the usefulness of this framework and to assess its computational feasibility. The results so far are promising. The paper concludes with a discussion of future directions.

1 INTRODUCTION, MOTIVATION AND RELATED WORK

1.1 Introduction and motivation

Traditionally, knowledge bases in knowledge based systems only include representations of knowledge from one person, usually an expert in some field such as medicine, law or chemistry. When knowledge bases are built to represent multiple experts' knowledge, the knowledge is almost always constrained to be consistent. Techniques such as the Delphi Method are used during knowledge acquisition, to iron-out inconsistencies and disagreements between the experts, but only the resulting consensus viewpoint is explicitly represented in the knowledge base to be used for inference. *The over-riding tendency in these methodologies is to see conflict as a problem.* It is interesting to note that the domains in which traditional expert systems are most successful are those in which there is some notion of `objective truth', legislation or received wisdom which minimises the possibility of conflict.

The purpose of this paper is to explore whether we can instead view conflict as an *opportunity* for interesting reasoning.

In the real world, absolute consensus or consistency is very rare. People have their own opinions and differing viewpoints, and as a result people commonly disagree with one another. A great deal of intelligent activity - discussions and arguments, learning and teaching - could not occur if the agents involved did not have differing viewpoints.

A particular example of this which has inspired the research described here, is in natural resource management. When decisions have to be made which involve changes to natural resources such as oceans, forests or the atmosphere, the interests of various parties such as land-owners, environmental pressure groups, wildlife biologists, governments or industries, need to be weighed up. Industries are currently under increasing pressure to carry out environmental impact assessments of their proposed development plans, and to integrate their results into their decision making processes. However, ecosystems are highly complex phenomena and often a range of relevant specialists will produce conflicting predictions of the effects of environmental changes. There are good reasons for this, the principle one being that it is impossible to construct models of ecosystems which are in any sense *complete*. It is essential to use simplifying assumptions, or *idealisations*, in order to construct tractable models or to stand any chance of verifying them with data collected in a feasible time frame. (See [Haggith 90] for more discussion of this issue). In addition, conflicts can arise due to differing interests or goals, for example, conservation of particular wildlife habitats versus economic development. Conflicts can also result from the use of different terminologies by people from different disciplines - such conflicts are sometimes hard to untangle as both agreement and disagreement can be shrouded by differing language use.

This presents a problem in the development of knowledge based decision support. Single, consistent knowledge bases are incapable of fulfilling the needs of situations such as these. Instead we require systems containing multiple knowledge bases, each of which can represent a particular viewpoint on a problem. We then need mechanisms for identifying, exploring and evaluating conflicts within and between knowledge bases.

The overall aim of the research described in this paper is to construct a knowledge based system in which conflicts can occur in knowledge bases, and which provides support for their exploration. The role of the user should be less like a novice asking a question of an expert who knows the answer, and more like a responsible decision-maker who calls, and chairs, a meeting of various relevant experts or interested parties. These agents can each argue for various positions, and the debate can be managed so that relevant counter-arguments are presented at the appropriate times. By interacting with the system in this way, the human user becomes better informed and more aware of the nature of the choices she has to make in order to make a wise decision. This model of system-user interaction accords with the ethical need for a human *locus of responsibility*, as explained in [Whitby 88].

1.2 Related Work

In this section some reasons for the negative view of conflict are examined and some pointers are provided to related research which challenges this view. The main area of related research concerns *argumentation.*

There have been few attempts to capture arguments in KBS applied to natural resources, though [Robertson et al 91] use arguments to aid in the construction of ecological simulation models, and [Robertson & Goldsborough 94] discuss representation of arguments for park site selection in Prolog.

Most argumentation research is done in the domain of law. There is considerable interest in systems such as IBIS (Issue Based Information System) which allow legal regulations to be navigated around, and redrafted, using high-level relations between sections of text which indicate, for example, that one regulation subsumes another. Some examples are [Conklin & Begeman 88] and [Casson & Stone 92]. An important limitation of this work is its informality - the relations are not logically constrained, nor intended to have a particular formal interpretation, which provides the systems with great flexibility, but limits the amount of reasoning which can be automated using these relations. A similar criticism applies to other, linguistic, approaches to the mark-up of texts, notably the Rhetorical Structure Theory of [Mann et al 89]. Within the domain of law there is other work of a more formal nature, such as the resource-bounded argumentation strategies of [Loui et al 93] and the representation of legal arguments as logic programs by [Kowalksi & Sergot 90] and [Sartor 93], who addresses the problem of conflict between laws. Earlier work on formal argumentation in law includes [Toulmin 56].

Toulmin's ideas have been used as a basis of argumentation work in other domains, such as medicine. The algorithmic decision procedures of the early medical expert systems are giving way to a less prescriptive approach of supporting the formation of arguments for different diagnoses, and weighing up the alternatives. Examples are [Fox & Krause 92], [Fox et al 92], [Krause et al 93]. In this work, and that of [Sartor 93], conflicts are resolved by providing *orderings* of either propositions or argument structures. In this paper no attempt at all is made to resolve disagreements - the role of the computer here is to store and articulate disagreements rather than to impose consistency. A similar ideological stance towards inconsistency can be found in [Rescher & Brandom 79] and in [Gabbay & Hunter 91], [Gabbay & Hunter 93a] and this latter approach is being applied in practice to supporting distributed software engineering by [Finkelstein et al 93].

In order to explain why a meta-level framework is considered a useful way to study disagreement, we need to look more closely at the issue of inconsistency. The reason that inconsistency is usually perceived as a problem is because most knowledge based systems are founded in classical logic. For example, a rule based system using forward chaining, can be viewed as simply an implementation of the application of modus ponens in a restricted form of first order predicate logic. In classical logic, there is a postulate called 'Ex Falsio Quod Libet', which states that anything can be deduced from a contradiction or falsity. Thus if we allow inconsistencies to occur in our knowledge bases, we effectively have 'logical chaos' as all statements are deducible. However, if we really want our knowledge bases to reflect the differing opinions of multiple experts, they will inevitably contain inconsistent statements.

There are three possible ways out of this impasse. The first is to allow the knowledge base to contain inconsistencies but to limit the set of possible inference rules in the system, and in particular to ban Ex Falsio Quod Libet. This is the **paraconsistent** approach of [Da Costa 74]. For an excellent survey of paraconsistency see [Priest et al 89].

The second approach is again to construct inconsistent knowledge bases but to limit the set of propositions which inference rules can be applied to. This is the **restricted access** approach of [Gabbay and Hunter 93b].

The third approach is the **meta-level** approach adopted here and also used in the field of truth maintenance systems [Doyle 79], [de Kleer 86], in some work on belief revision [Gardenfors 92], [Doyle 92] and in [Gabbay & Hunter 93a] (which is rare in not insisting that the purpose of the enterprise is to restore consistency to a belief set.) In the meta-level approach, no attempt is made to create a single knowledge base in which inconsistency is handled, and reasoned *with* in the object-level logic, but rather, the inconsistencies between knowledge bases are reasoned *about* at the meta-level. The meta-level approach has a significant benefit which is that it allows us to abstract away from any particular object-level knowledge representation formalism or set of inference techniques. A meta-level framework for reasoning about differences of opinion, and for reasoning about the arguments which are used to support those opinions, can be applied to any object-level knowledge representation system. It is thus general purpose. More significantly still, it provides the first stage of a methodology for reasoning about multiple knowledge based systems which may come from different sources, and each use their own representation language.

2 THE META-LEVEL FRAMEWORK

In this section, the formal language for reasoning about propositions and arguments is presented.

2.1 Meta-Level Objects

There are three categories of meta-level objects which the language describes:

1. Proposition Names
2. Arguments
3. Sets

2.2 Alphabet

The symbols used by the language are :

Ai ...Aj, Bi....Bj, to denote proposition names
<=, the argument constructor
{, }, $\cup$ (union), and $\in$ (member), as the set symbols
Some relation names (disagree etc) plus brackets and commas
&, v, ->, not, and the quantifiers, ie: the usual logical symbols

2.3 Term and Formulae Constructor Definitions

Terms in the meta-language are constructed by three methods.

1. All proposition names are terms.

2. Sets of proposition names are terms which are constructed by set construction :

> {} is a set (the null, or empty set)
> If T is a term, then {T} is a set.
> If S1 is the set {T1...Ti} and S2 is the set {Tj....Tn}
> then S1 ∪ S2 is the set {T1...Ti, Tj.....Tn}

3. Arguments are terms which are are constructed by argument construction :

> If P is a proposition name and S is a set of proposition names
> then P <= S is an argument.
> If P, Q are proposition names and P <= S, Q <= R are arguments,
> and S = S1 ∪ {Q}, then P <= S1 ∪ {Q <= R} is an argument.

The well formed formulae (wffs) of the language are defined as follows:

1. If T1..Tn are terms and Rn is an n-place relation then Rn(T1,....Tn) is a wff.
(Note that if R is a 2-place relation this will sometimes be shortened to T1 R T2.)

2. Wffs can also be constructed in the usual way using the logical connectives.

2.4 Destructor definitions

There are two destructor relations for looking inside argument and set terms.

1. Set membership, ∈, which is a two-place, infix relation.
Definition : P ∈ S if, for some S1, S = S1 ∪ {P}

2. Argument, which is a three-place relation.

> *Definition* : argument(A, P, S) if A is the argument P <= S.
> P is called the *conclusion* and S is called the *premise set*.

A further related destructor can be defined here from the other two. This is the *support* relation which holds between two propositions, the second of which occurs in an argument for the first.

> *Definition* : supports(P,Q) holds if there exists A, argument(A,P,S) and,
> either Q is a member of S, or there exists A1, a member of S,
> such that argument(A1,P1,S1) and supports(P1, Q).

2.5 Primitive relations

There are four basic relations at the meta-level which express links and contrasts between object-level propositions. They are independent of any particular object-level interpretation, though they will generally reflect object level relations which can be defined quite straightforwardly.

1. *equivalent(P, Q)* : P and Q are names of propositions which mean the same, in terms of some function for transforming propositions in one object-level language into propositions in another language, or in terms of an object-level equivalence relation, or according to the opinion of some person marking up text.

2. *disagreement(P,Q)* : P and Q are the names of propositions which disagree or express a conflict, according to some object-level notion such as negation, or temporal inconsistency, or according to the opinion of some person marking up text.

3. *elaboration(P, S)* : P is a proposition name and S is a set of names of propositions which elaborate or embellish upon P, according to multiple instantiations at the object level, or according to the opinion of some person marking up text.

4. *justification(P, S)* : P is a proposition name and S is a set of names of propositions which are a justification of P according to some object-level justification procedure such as modus ponens, abduction, or some other justification such as source or evidence, in the opinion of some person marking up text.

Notice that to reduce such complex notions as justification to two place relations is an extreme abstraction. A great deal of information is thrown away by doing so. In an extension to this framework, each of these four relations could in fact be three-place relations, where the third argument takes the name, or some representation, of the object-level reasoning step. For example, *disagree(P, not(P), negation)*, or *justification(P, {Q, P->Q}, abduction)* etc.

This would require the addition of a new meta-level object type, such as 'name of reasoning technique' in addition to names of propositions, arguments and sets. See below, in section 2.7, for some indication of how this extra meta-level information might be used. Note also, that information about the reasoner, knowledge base, agent or user could also be retained in the same way, with a corresponding cost of complexity.

2.6 Meta-level rules

It is possible to provide some properties of the above relations which constrain their possible applicability. This hence provides a test which can indicate which relations in an object-level language are suitable candidates.

Here are some examples :

1. *disagree(P, Q) -> disagree(Q, P)*
   (ie: disagreement is symmetrical)

2. *equivalent(P, Q) -> equivalent(Q, P)*
   (ie: equivalence is symmetrical)

3. *disagree(P, Q) -> not(equivalent(P,Q))*

4. *equivalent(P,Q) -> not(disagree(P, Q))*
(ie: disagreement and equivalence are mutually exclusive)

5. *(disagree(P,Q) & equivalent(Q, R)) -> disagree(P,R)*
(ie: disagreement is 'contagious' within equivalence sets)

6. *(equivalent(P,Q) & elaboration(P, S)) -> elaboration(Q, S)*
(ie: elaborations apply to all members of equivalence sets)

7. *elaboration(P, S) -> (Q e S, not(disagree(P, Q)))*
(ie: elaborations must not disagree with the statement they elaborate on)

8. *justification(P, S) -> (Q e S, not(disagree(P, Q)))*
(ie: justifications for a statement must not disagree with it).

2.7 Actual arguments

The definition of argument construction above (statement 3 in section 2.3) was purely syntactic, and actually defines the set of all possible argument forms, not all of which are interesting arguments. The subset of interesting arguments are those which describe actual object-level justifications, in other words where the premisses are actually used in some knowledge base to justify the conclusion. A further definition can thus be given as follows :

Definition : If P is a proposition name, S is a set of proposition names and *justification(P,S)*, then P <= S is an *actual* argument.
If P, Q are proposition names and P <= S, Q <= R are actual arguments, and S = S1»{Q}, then P <= S1 » {Q <= R} is an *actual* argument.

Arguments can be further subdivided by type if extra information about the sort of justification is also recorded as a third argument in the primitive relations, as suggested in section 2.5. There could thus be other argument definitions such as logical argument, invalid argument, appeals to authority, *ad hominem* argument and so on, which would in effect enable categorisation of arguments into a taxonomy of argument types. This kind of treatment could be valuable in a debating system.

One definition which can be given using only the syntactic argument form is as follows. It captures the notion of an argument bottoming out in axioms, or assumptions. In an argument P <= S, let P be called the conclusion and S the set of premisses.

Definition : An argument is *complete* if its set of premisses is empty, or if all members of the set of premisses are themselves complete arguments.

To capture the notion of creating arguments by putting heads together, ie: using a combination of reasoning from different object-level systems, we need to introduce substitution of a proposition in an argument, by one that is equivalent to it. Thus, if *supports(P,Q)* in an argument A, and *equivalent(Q,R)* we can generate a new argument by substituting an argument for R into A in place of the argument for Q. The same can be done for elaborations. These arguments are called *hybrid* arguments.

2.8 Mark-up versus Generation

An important aspect of the framework given above is that it can be used in two different ways, either as an *abstraction* from existing knowledge bases, or as a *specification* of the contents of knowledge bases which are yet to be constructed. In the first case, a process of abstraction from the object-level language to the meta-level language is required. In some object-level languages, such as first order predicate logic and its relatives, this process can be automated. In other languages, a process of "mark-up" is required by which the propositions are picked out and the argument structures described using the meta-level relations. An example of mark-up is given below, in which the object-level language is English text. It is one of the strengths of this framework that it may enable conflicts between formal and informal (eg: hypertext) knowledge bases to be reasoned about.

In the second case, specification, the meta-language can provide a way of articulating the arguments which will need to be generated by knowledge based systems, independently of any decision about the particular representation language of these knowledge bases. This can provide a means of capturing the *requirements* for a knowledge based system's contents and may prove useful in ensuring from the start that important stake-holders' views will be incorporated into a system and also a way of sketching at a high-level what sort of reasoning the system will be required to carry out in order to facilitate selection of an appropriate inference technique.

3 AN EXAMPLE TO ILLUSTRATE THE FRAMEWORK

3.1 Introduction

This section illustrates the meta-level argumentation language, by taking two excerpts from interview transcripts and showing how they might be marked-up to produce abstract representations of the arguments within them. These abstract representations consist of statements in the meta-language. To illustrate the usefulness of working at this abstract level, a comparison is made with a possible object-level representation of the content of the abstracts (a restricted, horn clause, form of first order predicate calculus). This is done to point out how difficult it is to carry out such a formalisation without knowing in advance which inferences will need to be possible. Once formalised it is difficult to see the overall form of the arguments, or to focus on the crux disagreement upon which a counterargument turns - there is a sense of not being able to see the wood for the trees. The meta-language provides a level of representation appropriate for this sort of overview.

This section contains an example in which there is more than one interpretation of the transcript text, where a counter-argument could turn on either of two disagreements. It is straightforward to mark-up these two alternatives in the meta-language but the repercussions of the two alternatives in the object-level are significant, and most importantly, they are very difficult to predict completely. In section 4.3 we will see how the meta-level argument structures can help to inform the decision between the two alternatives. The point here is that the meta-language is not intended as a *replacement* for more conventional object-level representation languages, but rather as a helpful addition which could aid in the design, maintenance or inspection of knowledge bases. It provides an explicit sketch of argument structures which are otherwise *implicit* in the knowledge base which can be useful where those arguments are liable to changes or need to take into account various points of view.

The two excerpts are from interviews of lay people, about their opinions of the likely effects of the greenhouse effect on the climate of Great Britain (a topic chosen for its high incidence of controversy and uncertainty). More details of the knowledge elicitation process can be found in [Haggith 93]. The first excerpt comes from an interview with an informant called Shen, the second with James T. Kirk (informants chose their own names). They contain conflicting predictions of sea-level rises occurring due to the melting of the ice-caps resulting from the greenhouse effect. Some interspersed comments on other related issues have been deleted from the second excerpt to keep the example as simple as possible. The capitalised text is the interviewer, the lower case text is the informant in each case.

3.2 Excerpt 1

The first excerpt is as follows :

> And there's talk about how the polar ice-caps will melt, and how Edinburgh will be submerged in a foot of water within the end of the, whatever, next century or something, I don't know. Something like that though.
> OK, WHY IS THERE A CONNECTION THERE? WHY IS THERE A CONNECTION BETWEEN SUBMERGING EDINBURGH AND ICE-CAPS?
> Well because it's mostly ice and when it becomes water then the level of the sea will rise, and we're quite near the sea here.

Mark-up :

Mark-up of the transcripts consists of identifying the propositions or statements in the text (this is usually referred to as *segmentation*), and the meta-level relations between them. Justification relations can then be used to construct arguments in the meta-language.

Propositions :

A1 : (There’s talk about how) the polar ice-caps will melt.
A2 : Edinburgh will be submerged in a foot of water.
A3 : Its mostly ice [presumably meaning the polar ice-caps].
A4 : When it [the ice] becomes water then the level of the sea will rise.
A5 : Were quite near the sea here [in Edinburgh].
A6 : The level of the sea will rise [conclusion of A4].

If we are using an object-level representation language such as horn clause logic, we need to formalise the contents of each of these sentences. This could be done as follows, using predicate names to represent relations or processes, with arguments representing objects, reserving the first argument for a crude representation of the temporal nature of the predicate. Constants begin with lower case letters, and variables (all considered universally quantified) are in upper case :

A1 : *melt(future, icecaps).*
A2 : *under(future, edinburgh, sea).*
A3 : *contains(always, icecaps, ice).*

A4 : *melt(T, ice) -> increase(T, attribute(sea, level)).*
A5 : *near(always, edinburgh, sea).*
A6 : *increase(future, attribute(sea, level)).*

We can now introduce some meta-level relations indicating the reasoning steps and use them to generate arguments.

Relations :

R1 : justification(A2, {A5, A6})
R2 : justification(A6, {A1, A3, A4})

To capture these at the object level we need to do some extra work. For R1 to hold we need some sort of general inference principle or *warrant* [Toulmin 56] explaining how a sea level rise affects areas near the sea, such as :

W1 : *(near(T, X, sea) & increase(T, attribute(sea, level))) -> under(T, X, sea).*

We also need to be able to do some sort of sensible matching of the temporal constants. For the second justification, R2, we need a warrant which connects the melting of an object to the melting of its contents, such as :

W2 : *(melt(T, X) & contains(T, X, Y)) -> melt(T, Y)).*

Note that we have had to make some significant assumptions at the object level about the type of inference which is possible. The warrants enable the two justifications to follow from the propositions by the application of modus ponens, together with a particular form of matching of temporal constants. At the meta-level we have not had to make any such specific assumptions, and the claim here is that it is helpful to be able to defer decisions such as this until we have a clearer picture of the knowledge base as a whole.

Arguments :

From R1 and R2 we can construct the following argument

Arg1 : A2 <= {A5, A6 <= {A1, A3, A4}}

[In fact the complete set of argument is as follows :
A2 <= {A5, A6}
A5 <= {}
A6 <= {A1, A3, A4}
A1 <= {}
A3 <= {}
A4 <= {}
A2 <= {A5 <= {}, A6 <= {A1<= {}, A3<= {}, A4<= {}}}]

3.3 Excerpt 2

This second excerpt expresses a different point of view on the issue of sea-level rise.

WILL THERE BE ANY CHANGE IN SEA LEVEL?

> Mm... I don't think there'll be anywhere near the amount people make out, ...   the ice melting doesn't matter.
> WHY?
> Because the majority of ice melt will be at the Arctic and that's ice that sits on the sea and displaces the water anyway. The ice melts, all that volume is replaced by water. Nine-tenths of ice sits below the surface of the water, and its density is lower.

Mark-up :

Propositions :

B1 : I don't think there'll be anywhere near the amount [of sea-level change] people make out.
B2 : The ice melting doesn’t matter [does not affect sea-level]
B3 : The majority of ice-melt will be at the Arctic.
B4 : Arctic ice sits on the sea and displaces the water.
B5 : The ice will melt.
B6 : When the ice melts, [then] all that volume is replaced by water.
B7 : Nine-tenths of the ice sits below the surface of the water.
B8 : Ice is less dense than water.

In this instance the location of objects (the ice caps in particular) is important to the reasoning and we will thus include a final argument in some predicates to represent the location in the object-level representation. In fact this argument is considerably more involved than the first one and will require much more work to formalise. A possible initial formalisation is as follows :

B1 : *not(change (future, attribute(sea, level)).*
B2 : *melt(T, ice) -> not(change(T, attribute(sea, level))).*
B3 : *melt(future, ice, arctic).*
B4 : *on(always, ice, sea, arctic) & displace(always, ice, sea, arctic).*
B5 : *melt(future, ice).*
B6 : *on(T, ice, water) & melt(T, ice) -> replace(T, attribute(ice, volume), attribute(water, volume)).*
B7 : *below(always, ice, water, 90%).*
B8 : *less_than(always, attribute(ice, density), attribute(water, density)).*

The formalisation is carried out in a haphazard manner at this stage because precisely what interpretation of the sentences is most appropriate depends on what inferences will need to be drawn at the object level. However, by adding some required meta-level relations we can see what sort of inferences should be possible at the object level and later attempt to adjust our formalisation accordingly. Assume that the following relations are marked-up.

Relations :

R3 : justification(B1, {B2, B5})
R4 : elaboration(B2, {B3, B4, B6})
R5 : justification(B6, {B7, B8})

Arguments :

Arg2 : B1 <= {B5, B2}
Arg3 : B6 <= {B7, B8}

To enable the corresponding object-level inferences to go through we need general warrants again, such as those relating volume to density, those explaining the effects of floating ice melting and the fact that the sea consists of water. Such warrants will not be formalised here as it is sufficient to point out that the task of enabling the correct object-level inference is a non-trivial task and that we could easily get bogged down in low-level details, such as the relations between the different arity *melt* predicates, the use of temporal predicates and so forth. The broad strokes of the meta-level language, on the other hand, allow us to move swiftly onto comparison of the two points of view.

Inter-transcript relations :

To bring out the structure of the counter-argument presented in excerpt 2 we need to look at relations between the excerpts. The following relations can also be marked-up :

R6 : equivalent(A1, B5)
R7 : disagree(A6, B1)
R8 : disagree(A4, B6)

That is, the two informants agree that the ice-caps will melt, but disagree on whether this will result in a rise in sea-level. They also disagree about what happens when the ice-melts. This is expressed by the relation R8.

Problem : The counter-argument could, however, be expressed by a different disagreement, between A4 and B2 :

R8a : disagree(A4, B2).

Precisely where the disagreement occurs affects the structure of the counter-argument, as will be seen in section 4.3. The way we have represented this knowledge at the object level makes it appear more likely that B2 is the source of disagreement with A4, rather than B6, as B2 and A4 have identical antecedents, and contradictory consequents. This can be seen by inspection :

A4 : *melt(T, ice) -> increase(T, attribute(sea, level)).*
B2 : *melt(T, ice) -> not(change(T, attribute(sea, level))).*
B6 : *on(T, ice, water) & melt(T, ice) ->*
replace(T, attribute(ice, volume), attribute(water, volume)).

B6 is syntactically much less close to A4, so if we wish the disagreement to hold between them, then significant reformalisation will be necessary.

Two questions arise now. Firstly, how should we decide which of the two possibilities is more appropriate? Secondly, what changes need to be made to the object-level representation in order to make the disagreement explicit so that it can be detected as a contradiction in the object-level logic?

It should hopefully be clear that there are several possible answers to the second question any of which would affect the possible inferences at the object-level. For example, if we alter the *change* predicate to *increase*, in B2, we have a clear contradiction. However we will also need to alter B1 in order to continue to be able to deduce it from B2 and B5. We run the risk of tinkering with our object level representation in an unprincipled manner.

The remainder of this paper seeks to show how working at the meta-level will enable us to build up a clearer overview of the reasoning patterns made possible by the stated relations, allowing us to decide between alternative relations and thus answering our first question. Having selected one option we will then have sufficient understanding of the required relations between the propositions to be able to make the minimal alterations to the object-level representation to enable the chosen inferences and contradictions, whilst interfering as little as possible with inferences which are currently possible.

4 USING THE META-LANGUAGE TO EXPLORE CONFLICTS

4.1 Introduction

This section sketches an implementation of the meta-level framework as described in section 2. A simple prototype has been constructed in Prolog to allow the following four basic facilities :

(1) Creation of meta-level objects,
(2) Mark-up of relations between them,
(3) Exploration of conflicts,
(4) Construction of higher-order argumentation structures.

The first two facilities are very simple in the Prolog implementation, so a hypertext tool has also been constructed (using HyperCard [Goodman 90]) which allows a piece of text to be edited and segmented into propositions, and allows relations between propositions and sets of propositions to be marked-up and written to a file which can be read by the Prolog system. Facilities (3) and (4) are much more interesting and will be described below.

4.2 Conflict exploration

In order to define a system for allowing a user to interact with the objects in the framework, the concept of a **conflict set** is useful. This set is a subset of all the proposition names, and it is used to represent those propositions which a user is currently interested in. Typically it will contain at least two propositions which are in disagreement, plus other statements equivalent to, or elaborating on those statements. It can also contain justifications for the propositions which may then generate further points of disagreement. The conflict set can be thought of as a window on the controversies within the knowledge base which can move around according to the relations between propositions. Thus exploration is not confined to moving according to relations or structures linking propositions at the object-level.

Conflict exploration results from altering the conflict set, by either *expanding* it to include new propositions related to those already in the conflict set, or *focussing* it to eliminate those propositions which are not interesting or which have been sufficiently explored. There are various different ways in which expansion and focussing can occur. These are described next.

Expansion of conflict set

Here are four simple methods for expanding the conflict set.

1. *Adding support* : take a proposition in the conflict set, find a justification relation for it, and add the justifications to the conflict set.

2. *Adding disagreement* : take a proposition in the conflict set, find a disagreement relation involving it, and add the conflicting proposition to the conflict set.

3. *Substituting equivalents* : take a proposition in the conflict set, find another proposition equivalent to it and replace the first by the second in the conflict set.

4. *Substituting elaborations* : take a proposition in the conflict set, find an elaboration of it and replace the proposition with the set of elaborations.

The following example illustrates these four exploration moves.

Assume the conflict set starts out containing the conflict between P6 and Q1 in the previous section, thus:

A6 : The level of the sea will rise
B1 : I don't think there'll be anywhere near the amount [of sea-level change] people make out.

Method 1 can add justifications for either or both of these. The justification for A6 is the set of A1, A3 and A4. Adding them gives the following new conflict set :

A6 : The level of the sea will rise
A1 : (There's talk about how) the polar ice-caps will melt.
A3 : Its mostly ice [meaning the polar ice-caps].
A4 : When it [the ice] becomes water then the level of the sea will rise.
B1 : I don't think there'll be anywhere near the amount [of sea-level change] people make out.

Method 2 can now use the disagreement (R8) between A4 and B6 to add B6 :

B6 : When the ice melts, [then] all that volume is replaced by water.

Method 3 can add B5 using the equivalence statement R6.

B5 : The ice will melt.

Method 1 can add B5 and B2 as justifications of B1, so this will result in the addition of just B2 as B5 is already in the conflict set. (Note that if the disagreement had been stated as between A4 and B2, as discussed in section 3, then B2 could have been added using method 2.)

B2 : The ice melting doesn't matter [does not affect sea-level]

Finally Method 4 can add B3 and B4 as elaborations of B2 (B6 is already present).

B3 : The majority of ice-melt will be at the Arctic.
B4 : Arctic ice sits on the sea and displaces the water.

Following these five exploration moves the conflict set has grown to be as follows :

A6 : The level of the sea will rise
A1 : (There's talk about how) the polar ice-caps will melt.
A3 : Its mostly ice [meaning the polar ice-caps].
A4 : When it [the ice] becomes water then the level of the sea will rise.
B1 : I don't think there'll be anywhere near the amount [of sea-level change] people make out.
B6 : When the ice melts, [then] all that volume is replaced by water.
B5 : The ice will melt.
B2 : The ice melting doesn't matter [does not affect sea-level]
B3 : The majority of ice-melt will be at the Arctic.
B4 : Arctic ice sits on the sea and displaces the water.

This is clearly becoming unwieldy so some means of refocussing it is required.

Focussing of conflict set

Here are two simple methods for focussing the conflict set.

1. *Removal of equivalents* : if the conflict set contains two or more propositions which are equivalent to each other, remove all but one of them.

2. *Removal of uncontroversial propositions* : if any proposition in the conflict set is not in disagreement with any other proposition in the knowledge bases, remove it from the conflict set.

Continuing the example from above, the first method would remove either A1 or B5, and the second method would remove all statements except the following four which represent the explicit disagreements in the knowledge base.

A6 : The level of the sea will rise.
B1 : I don't think there'll be anywhere near the amount [of sea-level rise] people make out.
B6 : When the ice melts, [then] all that volume is replaced by water.
A4 : When it [the ice] becomes water then the level of the sea will rise.

Expansion could now continue in a different direction, for example, by following the justifications for B6.

All of these methods of exploration have been implemented in the prototype. Although the current interface is very crude, the basic idea of a proposition set expanding and contracting is clear and seems fairly intuitive. It can be tentatively concluded that this is a useful technique for revealing the connectivity implicit in a knowledge base, and for simulating the 'flow' of an argument or debate.

The prototype has also suggested some limitations of the original ideas which will feed into the next stage of implementation. One is that it is important to supply a user with grounds for choosing a particular exploration move - namely which propositions are closely related to those currently in the conflict set. The user currently selects exploration moves 'blind' in the sense that they have no way of knowing how the conflict set will change until after the exploration move is completed. One way of handling this would be to allow the user to see the 'corona' of the conflict set - ie: all those propositions which are directly related to its members. Broadening the horizon of the conflict set in this way will also facilitate mark-up as the user will be able to notice where relationships which should hold are missing from the meta-level representation. More sophisticated 'browsing' facilities would also help here. An alternative (or addition) to this extension, would be automatically to suggest interesting exploration moves which a user might like to take, using some sort of strategy such as first suggesting propositions which disagree with one of the conflict set, then justifications and so forth. Some of the ideas about more complex argumentation structures in the next section could also support more intricate exploration.

4.3 Synthesis of argumentation structures

The previous section described how the four basic relations could be used to allow navigation around, or exploration of, marked-up knowledge bases containing conflicts. Another approach to the examples is to construct more sophisticated, or higher-order, meta-level relations defined in terms of the four primitive ones. Some examples follow to give a feel for what this would involve. They have all been implemented.

The first two and the last examples (A, B & F) were chosen because they are each a primitive relation in other peoples work on argumentation. The first, rebuttal, is a primitive in some current work on dialectical reasoning by Morten Elvang-Goransson. The second, undercutting, is a primitive in Toulmin's theory of argumentation. The last is from Sartor's work on representation of legal arguments. The fact that they can each be characterised in terms of the four basic relations I have chosen is encouraging.

A Rebuttal.

In [Elvang-Goransson et al 93], *rebuttal* is defined as a relation between arguments whose conclusions disagree. (Strictly speaking they say that the truth of the conclusion of one must imply the falsity of the conclusion of the other, but in my system this is one possible object-level interpretation of disagreement. The definition of rebuttal in terms of meta-level disagreement, has, I would claim, greater generality than their definition in terms of object-level implication and negation). The meta-level definition is as follows :

> *Definition :* *rebuttal(P)* is the set of arguments, *A,* such that :
> *disagreement(P,Q) & argument(A, Q, _)*
>
> *Definition :* *rebuts(A, B)* if *argument(A, P, _) & member(B, rebuttal(P)).*

In other words, the rebuttal of a proposition P is the set of arguments for any propositions which disagree with P. Two arguments rebut each other if one is a member of the rebuttal of the conclusion of the other.

We hence have the following cases of rebuttal :

rebuts(A6<={A3,A1,A4}, B1<={B5,B2})

This is the rebuttal of 'The level of the sea will rise' (A6) by the argument in the second transcript.

If we select the disagreement between A4 and B6 (relation R8), then we have a rebuttal for A4 :

rebuts(A4 <={}, B6 <={B7,B8})

If we select the disagreement between A4 and B2 (relation R8a), the rebuttal is less interesting :

rebuts(A4 <={}, B2 <= {})

B Undercutting.

In [Toulmin 56], another primitive is *undercuts*, which means the following : an argument A, with conclusion Q, undercuts an argument B if for some premiss P of B, P disagrees with Q. In other words, an argument undercuts another, if the first rebuts a premiss of
the second.

Definition : undercutting(P) is the set of arguments, *A* , such that
argument(_, P, S) & member(P1, S) & disagreement(P1, Q)
& argument(A, Q, _)
or, alternatively, using the definition of rebuttal above,
argument(_, P, S) & member(P1, S) & member(A, rebuttal(P1))

Definition : undercuts(A, B) if
argument(A, P, _) & member(B, undercutting(P)).

For example : undercuts(P2 <={P5, P6}, Q1 <={Q5, Q2})

ie: Argument Arg1 is undercut by argument Arg2, because of the disagreement between P6 and Q1.

Returning to the two alternative disagreements there are two possibilities here too :

undercuts(A6 <={A3, A1, A4}, B2<= {}) or
undercuts(A6 <={A3, A1, A4}, B6 <={B7,B8}).

Hence we have richer argument structures appearing if we choose the disagreement between A4 and B6. This is due to B6 being supported by a justification and there thus being an argument for it.

Four other argument structures have also been implemented and can be automatically generated from an appropriately marked-up knowledge base. These are defined as follows :

C Corroboration.

The corroboration for a proposition is defined as the set of arguments for propositions equivalent to it.

Definition : corroboration(P) is the set of arguments, A, such that
equivalent(P, Q) & argument(A, Q, _).

There are no corroborations in the current example, however, if either A1 or B5 had arguments for them then they would mutually corroborate each other.

D Enlargement.

The enlargement of P is the set of arguments for all elaborations of P.

Definition : enlargement(P) is the set of arguments, A, such that
elaboration(P, S) & member(E, S) & argument(A, E, _).

In the running example, B2 is elaborated on by B6, B3 and B4, thus we have :

enlargement(B2, {B3 <= {}, B4 <= {}, B6 <= {B7, B8}})

E Undermining.

The undermining of a proposition is defined as the set of arguments for propositions which disagree with an elaboration of P, in other words, the set of rebuttals of the elaborations of P.

Definition : undermining(P) is the set of arguments, A, such that
elaboration(P, S) & member(E, S) & member(A, rebuttal(E)).

In the example, if B6 is selected as disagreeing with A4 then, as B6 elaborates B2, this means that A4 undermines B2, in other words, the claim that melting ice will cause sea levels to rise undermines the claim that melting ice will cause no change to sea levels.

F Counter-argument.

This final structure is the richest of all, and is closely related to the definition given by [Sartor 93], in which a counter-argument to a proposition P is defined as an argument for the 'complement' (negation) of any proposition used anywhere in an argument for P. We could interpret this in two ways, which will each bring out different structures and throw light on our decision between alternative disagreements. The first interpretation is as follows :

Definition : counterargument(P) is an argument, A, such that
rebuts(P, A) or
argument(P, B, Premisses) & member(Q, Premisses)
& counterargument(Q, A).

This is a recursive extension of the rebuttal definition. Using this definition, if relation R8 holds, then the argument for B6 is a counterargument for both A2 and for A6. However, because there is no argument for B2, using the alternative disagreement relation, R8a, gives no counterarguments.

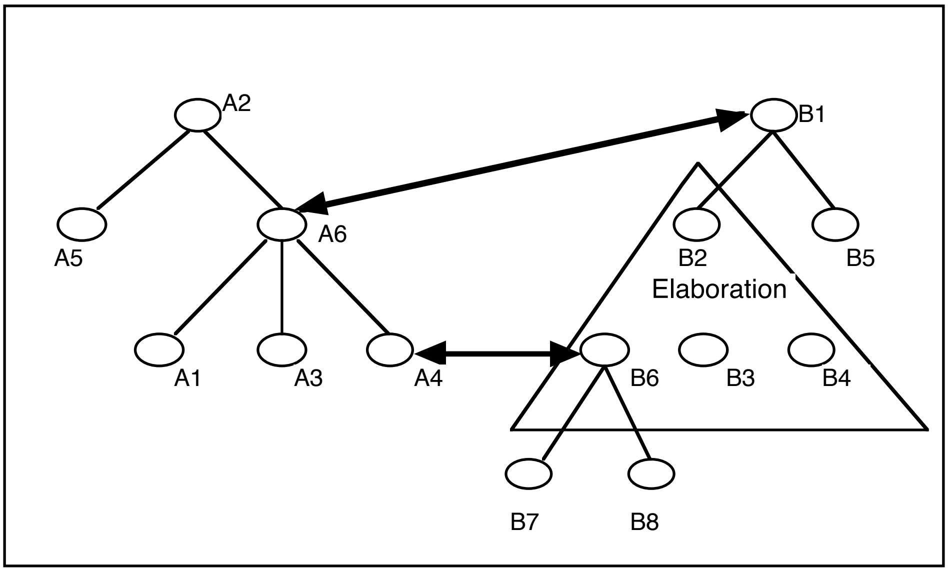
The secod interpretation would involve a richer definition of a counter-argument for P which would be an argument against any proposition used in any *hybrid* argument for

P (see section 2.7, where a hybrid argument is defined as an argument with equivalent and elaborated propositions substituted by their appropriate counterparts). There is a hybrid argument for B2, as follows :

Arg4 : B2 <= {B6 <= {B7, B8}, B3, B4}}

Under this definition of counter-argument Arg4 would be a counter-argument for both A2 and A6, and might be preferred because it has more structure than the argument for B6.

We have now formed a full picture of the structure of the arguments in our example, and this is summarised in the diagram below. The nodes in the diagram represent propositions, the thick arcs with double arrows indicate the disagreements and the other arcs represent justifications. The elaboration of B2 is indicated by the triangle.



We can see that it is preferable to state the disagreement between B6 and A4, rather than between B2 and A4 (as looked most likely at the object level), as the former will produce interesting argumentation structures, whereas the latter will not. We can also see why this is the case, which is that there is no argument for B2. We could adjust this by changing the elaboration of B2 into a justification, thus producing an argument for B2. Alternatively we can return to the object level and focus our attention on ensuring that B7 and B8 are represented such that they can be used to deduce B6, and perhaps also so that B6 and A4 are recognisably contradictory. We know also that changing the object-level representation of A4 might have repercussions for the deducibility of A6 and A2. Space does not allow the details at the object-level to be spelled out, and anyway the main point here is to point out that our work at the meta-level has enabled us to return to our object-level considerations with a clear idea of precisely what changes must be made and what side-effects must be watched for.

These structures provide a much broader view of the patterns of reasoning possible within a knowledge base, and provide a rich, coherently organised source of information relevant to any particular proposition. The current implementation can construct all of these sets for any proposition in its knowledge base. There are two obvious directions in which to take such an implementation. Firstly, as these are complex structures, it will be important to evaluate how costly the construction and maintenance is in computational terms. The current implementation is surprisingly capable of constructing them very quickly, but it remains to be seen how badly performance will degrade when reasoning over more substantial knowledge bases. Secondly, further examples from the argumentation literature could be implemented in a similar manner.

4.4 Knowledge base comparison

The six structures (arguments, corroborations, rebuttals, enlargements, undercuttings and underminings) could provide a user with detailed comparisons between knowledge bases, detailed in that they would provide, for each individual proposition, an analysis of how it relates to the other propositions in a knowledge base. The rebuttal, undercutting and undermining sets provide a detailed analysis of *conflicts*, so that those propositions with large sets are highly controversial. Propositions with large corroboration sets are points of *consensus*. Large argument sets and big enlargements are indications of highly *cohesive* knowledge bases. Likewise small argument sets or small enlargements indicate *fragmentary* knowledge. These structures could thus provide evaluations of the overall 'shape' of knowledge bases, either during their construction, which would require continuous updates to such structures for each proposition in the knowledge base, as a form of book-keeping, or to guide a user in exploration of completed knowledge bases. The usefulness of these structures in automating debate and providing decision support is currently being investigated.

5 CONCLUSIONS AND FUTURE DIRECTIONS

A meta-level framework for representing relations, particularly conflicts, in multiple knowledge bases has been described. An example has been given of how the meta-level language can be used either interactively, to help in exploration of conflicting knowledge bases, or automatically, to construct and reason about more complicated relationships between arguments. A prototype implementation of these ideas has been described to illustrate these two applications. This prototype is extremely encouraging as it has proved simple to implement (thus suggesting that the framework is computationally feasible) and successful in representing some simple examples (thus suggesting that the ideas are useful).

This research is still in its infancy and thus the prototype system has been an effective way of triggering new ideas for better ways of using the framework, as well as providing the means of a preliminary evaluation. These ideas for future work have been discussed in the paper as they have come up, but it seems worthwhile to summarise them here in order to conclude on an optimistic note.

As mentioned in sections 2.5 and 2.7, there are several possibilities for enriching the meta-level framework (making it less abstract) such as explicitly representing the object-level reasoning techniques (eg: modus ponens) or the agents whose knowledge is being used, as meta-level objects. This would produce a much more fine grained analysis which could be useful in evaluating, for example, *why* agents disagree. However, the less abstract the meta-level framework becomes, the greater the work needed in making a correspondence between meta-level objects and their object-level counterparts.

In section 2 there was also some discussion of the relative merits of seeing this framework as an *abstraction* of existing object-level knowledge bases (via a process known as mark-up) or as *specifications* of possible object-level knowledge bases (which could be created by a form of instantiation). It will be informative to assess each of these alternatives - the current paper makes no claims either way, although the running example involves the former approach - to see which is most applicable. One important question is whether it is possible to automate the mark-up process, and if so, what the constraints on this are. If not, the process of marking up knowledge bases 'by hand' could prove so expensive that the framework is not readily usable in this way. Likewise it will be interesting to assess the possibilities of using the framework for requirements capture, for example in the domain of safety critical systems, in which there is much interest in being able explicitly to capture the arguments, legislative constraints and expert opinions around system designs.

Finally, some suggestions were given in sections 4.2-4.4 as to how exploration could be enhanced by increasing the horizon of the conflict set, using the higher-order structures and providing automated guidance. These will now move to implementation and work will continue towards full-blown automated debate and its applicability to real world decision support.

REFERENCES

[Casson & Stone 92] Casson, A. and Stone, D. ‘An Expertext System for Building Standards’, in proceedings of a workshop on Computers and Building Standards, Montreal 1992.

[Conklin & Begeman 88] Conklin, J. and Begeman, M.L. ‘gIBIS : A Hypertext Tool for Exploratory Policy Discussion’ in *ACM Transactions on Office Information Systems*, Vol. 6, No. 4, October 1988.

[Da Costa 74] Da Costa, N.C.A. 'On the theory of Inconsistent Formal Systems', Notre Dame Journal of Formal Logic, Vol 15, No 4, pp 497-510, 1974.

[De Kleer 86] de Kleer, J. ‘An Assumption Based TMS’ *Artificial Intelligence*, Vol 28, 1986.

[Doyle 79] Doyle, J. ‘A truth maintenance system’.*Artificial Intelligence*, Vol 12, 1979.

[Doyle 92] Doyle, J. ‘Reason maintenance and belief revision : Foundations versus Coherence theories’, in [Gardenförs 92].

[Elvang-Goransson et al 93] Morten Elvang-Goransson, Paul Krause and John Fox, 'Dialectical reasoning with inconsistent information'. In proceedings of Uncertainty in AI 1993.

[Finkelstein et al 93] Finkelstein, A., Gabbay, D., Hunter, A., Kramer, J., and Nuseibeh, B. ‘Inconsistency Handling in Multi-Perspective Specifications’ in proceedings of Fourth European Software Engineering Conference (ESEC 93), Springer-Verlag, 1993.

[Fox & Krause 92] John Fox and Paul Krause, 'Qualitative frameworks for decision support : lessons from medicine' The Knowledge Engineering Review, Vol 7:1, 1992, pp19-23.

[Fox et al 92] John Fox, Paul Krause, Simon Ambler, 'Arguments, contradictions and practical reasoning' in proceedings of ECAI 92, 1992.

[Gabbay & Hunter 91] Gabbay, D. and Hunter, A. ‘Making Inconsistency Respectable : A Logical Framework for Inconsistency in Reasoning, Part 1 - A Position Paper’ in P. Jorrand & J. Kelemen (eds) *Fundamentals of Artificial Intelligence Research’* Lecture Notes in Computer Science 535, Springer Verlag, 1991.

[Gabbay & Hunter 93a] Gabbay, D. and Hunter, A. ‘Making Inconsistency Respectable : Part 2 - Meta-level handling of inconsistency’ in proceedings of ECSQUARU conference 1993, Lecture Notes in Computer Science 747, Springer Verlag, 1993.

[Gabbay and Hunter 93b] Gabbay, D. and Hunter, A. 'Restricted Access Logics for Inconsistent Information'. In proceedings of ECSQUARU conference 1993, Lecture Notes in Computer Science, 747, Springer Verlag, 1993.

[Gardenförs 92] Gardenförs, P. *Belief Revision.* Cambridge University Press, 1992.

[Goodman 90] Goodman, D. *The Complete HyperCard Handbook.* Bantam Books, 1990.

[Haggith 90] Haggith, M. 'Sheltering Under a Tree : an Artificial Intelligence approach to idealisation in ecological modelling'. In proceedings of Pacific Rim International Conference on Artificial Intelligence, Nagoya, Japan, 1990.

[Haggith 93] Haggith, M. 'Towards a formal analysis of disagreement', DAI Discussion Paper 128, Department of Artificial Intelligence, The University of Edinburgh, 1993.

[Kowalski & Sergot 90] Kowalski, R. A. and Sergot, M. J. ‘The use of logical models in legal problem solving”. *Ratio Juris,* Vol.3, No. 2. 1990.

[Krause et al 93] Krause,P., Ambler, S., Elvang-Goransson, M. and Fox, J. 'A Logic of Argumentation for Uncertain Reasoning', to appear in *Computational Intelligence,* submitted November 1993.

[Loui et al 93] Loui, R.P., Norman, J., Olson, J. and Merrill, A. 'A design for Reasoning with Policies, Precedents and Rationales', in the proceedings of the Fourth International Conference on Artificial Intelligence and Law, Amsterdam, The Netherlands, June 1993.

[Mann et al 89] Mann, W.C., Matthiessen, C.M.I.M. and Thompson, S.A. ‘Rhetorical Structure Theory and Text Analysis', ISI Research Report ISI/RR-89-242, University of Southern California, 1989.

[Priest et al 89] Priest, G., Routley, R. and Norman, J. editors. *Paraconsistent Logic : Essays on the Inconsistent.* Philosophia Verlag, 1989.

[Rescher & Brandom 79] Rescher, N. and Brandom, R., *The Logic of Inconsistency*, American Philosophical Quarterly Library of Philosophy, Vol 5, Basil Blackwell, Oxford, 1979.

[Robertson et al 91] Robertson, D., Bundy, A., Muetzelfeldt, R., Haggith, M. and Uschold, M. *Eco-Logic : Logic based approaches to ecological modelling*. MIT Press 1991.

[Robertson & Goldsborough 94] Robertson, D. and Goldsborough, D. 'Representing the Structure of Reserve Selection Arguments using Logic Programs'. Unpublished paper.

[Sartor 93] Sartor, G. 'A Simple Computational Model for Nonmonotonic and Adversarial Legal Reasoning', in the proceedings of the Fourth International Conference on Artificial Intelligence and Law, Amsterdam, The Netherlands, June 1993.

[Toulmin 56] S. Toulmin, *The Uses of Argument*. Cambridge University Press,1956.

[Whitby 88] Blay Whitby, *Artificial Intelligence : a handbook of professionalism.* Ellis Horwood Ltd., 1988.