Analytic Modal Revision for Multi-agent Systems

Paolo Di Giusto[1] Guido Governatori[2]

[1] Regione Emilia Romagna, viale Aldo Moro 52, 40122 Bologna, Italy,
e-mail coc0788@iperbole.bologna.it
[2] CIT, Griffith University, Nathan, QLD 4111, Australia
e-mail G.Governatori@cit.gu.edu.au

Abstract. We present two models of hierarchical structured multi-agents, and we describe how to obtain a modal knowledge base from distributed sources. We then propose a computationally oriented revision procedure for modal knowledge bases. This procedure is based on a labelled tableaux calculi supplemented with a formalism to record the dependencies of the formulae. The dependencies are then used to reconstruct the minimal inconsistent sets, and the sub-formulae responsible for the inconsistencies are revised according to well-defined chains of modal functions.

1 Introduction

Individuals are able to build a model of the world, and so are institutions. In a common (even if a little idealized) version of this process of model building, it is assumed that a knowledge base is usually built using pieces of information collected from "outside". The knowledge is only partial, and the process of acquiring data is unending. As data are acquired, they are also incorporated in theories: a knowledge base is not just a collection of facts, but also a system of rules connecting them. It is assumed that a set of rules is present from the beginning; new data are used for improving and refining it. It may happen that new (reliable) data are not fully compatible with the knowledge base, so we have to modify (revise) it to accommodate them. Moreover, data are acquired from several different sources, each one probing a sector of an environment (physical or conceptual). So, the process of building a knowledge base deals with distributed and partial data. As information collected from various sources may be contradictory (from one source we get p and from another source we get $\neg p$), using a modal language seems natural. In this way the incompatible pieces of information p and $\neg p$ are represented as $\Diamond p$ and $\Diamond\neg p$, meaning that both are possible.

In section 2 we present two hierarchical models of agents, then, in section 3 we show how to construct modal knowledge bases arising from the above models. In sections 4 and 5 we describe a revision procedure for modal knowledge bases, and in section 5.1 we propose a tableau formalism to be used in the process of revision.

2 Sensors, Agents, Supervisors

The Basic Model (SA-model) The basic model, or SA-model, includes two components: a set of sensors and one agent. Sensors perform measurements and send the results to the agent. For the sake of simplicity we assume that the sensors share the same language, consisting of a set of properties P. When a sensor s performs a measurement and finds that a property p holds in its location, then we say that p holds at s. If the property p does not hold, then we say that $\neg p$ holds.

The agent, on the other hand, has a richer language than the sensors. First, its language includes modal operators allowing the coordinatation of information sent by sensors. When the sensor s_i informs the agent that p holds at s_i then a fact $\mathrm{K}_i p$ is added to the agent's theory, meaning that the agent has received the piece of information p from the sensor s_i. Moreover the language is supplemented with the modal operator $\Box$ and its dual $\Diamond$, conceived of as *knowledge* operators. The meaning of $\Box$ and $\Diamond$ is

$$\Box p \text{ iff all sensors read } p \qquad \Diamond p \text{ iff there is at least one sensor which reads } p\ .$$

Second the agent is equipped with a number of rules about the world. This means that the agent, contrary to the sensors, has a theory of the world, which may include rules expressed in a modal language, using a suitable system of modal logic. The theory the agent starts with might not be the right one; it might be the case that the information sent by the sensors is not compatible with the agent's theory. In this case the theory must be changed in order to accommodate the new facts.

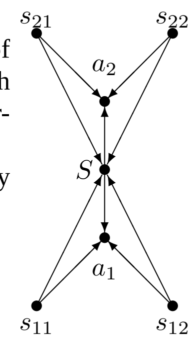
Different kinds of theory change have been developed, according to different basic assumptions about the status of old and new information. The simplest case is that in which the sensors are reliable and the world is fixed. In this case new data are accepted and will be never contradicted by further evidence. If the agent knowledge base is partitioned into defeasible and non-defeasible formulae, then the only defeasible formulae are the rules the Agent started with (this is the standard case of the AGM approach). Another case is that in which the sensors are reliable but the world can change. In this case (whose study was initiated by [11]) all formulae are defeasible except the last piece of information. When the sensors are not reliable some kind of preferential model is necessary. In these models, the (possibly contradictory) evidences coming from different sensors are weighted against a (pre-loaded or incrementally built) factor of confidence of the different sensors. In this way, evidence may also be rejected. We shall not pursue this line here, however.

A Hierarchical Model (SAS-model) The basic model described in the previous section treats the sensors as slaves: they do not have a theory about the world, do not perform inferences, do not have to revise theories. The Agent is the only intelligent component of the system. However, a more complex model can be built assuming that a set of agents are connected to a supervisor. We may think of autonomous robots, having a certain degree of intelligence, which are coordinated by a central computer. Each robot, in turn, unleashes a number of slave sensors to gather data about the world. The supervisor has its own theory about the world, and a subset of the theory is shared by all the agents. It is a higher level theory, which is then specialized by lower level theories at the Agent level. In this case, theory revision (or update) can occur at two levels and information might flow in both directions.

Different patterns of communication can be envisioned. The pattern we are concerned with is studied to minimize the interactions between levels: at each stage, only two (possibly non-adjacent) levels are involved. The difference between communicative events lies in the different status deserved by data sent by the sensors and by the supervisor.

Van Linder et al. [12] discussing a homogeneous environment, argue in favor of an individualistic concept: in their wording, *seeing is stronger than hearing*. In our structured, hierarchical environment, we would better explore the idea of a priority accorded to the communications downloaded from the central computer.

Structuring a complex environment in hierarchical layers is a common strategy in order to keep communication complexity low. Accordingly, the burden of revision (and maybe decision) can be distributed across the layers.



In the diagram beside we present a simple framework made of a supervisor S, which rules over two agents a_1 and a_2, and each agent a_i ($i = 1, 2$) has two sensors s_{i1} and s_{i2}. The arrows determine the direction in which data flow.

Agents and sensors behave as in the SA-model as long as they can, that is, they get through a cycle of events of the type:

1a Sensors read data;
2a Sensors send data to their agent(s);
3a Agents get data;
4a Agents revise their own theory against data sent by their sensors and the supervisor theory.

If an agent cannot restore overall consistency, it means that the supervisor theory itself is not consistent with the facts the agent has access to. In this case, the agent asks for the intervention of the supervisor. The supervisor collects the data from all agents and revises its own theory in order to restore consistency:

1b Supervisor gets data from the agents;
2b Supervisor revises its belief system;
3b Supervisor broadcasts the revised theory to the agents.

However, it is possible that the theory is inconsistent for the agent but not for the supervisor; so we split the supervisor's theory into two parts: the first consists of global rules, i.e., rules that are passed to all the agents; the second contains local rules, i.e., rules that hold only under particular circumstances, and are transmitted only to given agents. In an environment with n agents this is implemented by $n+1$ sets: the set of the rules and a set of exceptions for each agent. A set of exceptions contains the rules that do not hold for a given agent. The rules passed to an agent are all the rules minus the exceptions for that agent. Once the supervisor determines inconsistencies occurring at the agent's level but not globally, it adds the culprit rules to the respective sets of exceptions. Note that the revision cannot change a consistent overall theory (agent+supervisor) into an inconsistent one; at most, it can change an inconsistent overall theory into a consistent one.

The last step is on the agents' side. They might simply accept the new theory, but this could result in too weak overall theories, in case that, while the now revised supervised theory was in effect, the agents had revised their own theories beyond need. So, they restore their original theory, and then revise it against sensors' data and supervisor's theory.

1c Agents get the revised theory from supervisor;
2c Agents restore their original theories;
3c Agents revise their theory against data sent by their sensors and the Supervisor theory.

This implementation of this scheme depends on how beliefs are represented and revised by actors.

In order to make communication of rules (and rule modifications) feasible:

a) the knowledge state of an actor is represented by means of a finite set of rules;
b) the revision/update operation is computationally well-defined, and results in a finite set of rules;
c) the revision (or update) operations should be as local as possible, that is, the change affects only a limited number of rules.

Condition c) expresses another aspect of minimality, i.e., a minimal change of theory form, which complements the usual views about a minimal change of theory content.

When an agent passes its revised rules to the supervisor, the latter has to treat the modal operators as local to the agent, that is, it has to interpret them in the same way the agent interpreted the sensor's data. Agents at different levels of the hierarchy share the same logical structure, so that it is easy to extend the hierarchy through other levels. However, the knowledge states of actors at different levels differ. The fact that central rules are down-loaded to lower-level actors need not mean that all central rules are down-loaded. We simply say that relevant rules are exchanged. The simplest relevance criterion is syntactic: only a subset of the central rules are relevant to each lower-level actor. In turn, the choice might reflect a lexical criterion: each lower-level agent is aware of only a subset of the features of the world, and so can access only a subset of the vocabulary (atomic constants and predicate symbols). This just means that the supervisor has to account for the different fields of action of the agents.

3 Modalities for a Multi-agent System

Two main reasons suggest the use of modal logic for the multi-agents framework we propose: 1) the epistemic interpretation of the notions involved; 2) data are gathered from different sources, that can be conceived of as possible worlds. In the models two kinds of modalities are involved: K_{ij} for a sensor s_{ij} and $\Box$ (and its dual $\Diamond$) for the agents and the supervisor.

In the SA-model we have an agent receiving data from n sensors, while in the SAS-model we have a unique supervisor supervising n agents. As in the SA-model each agent a_i, $(1 \leq i \leq n)$ has i_m sensors. Both models share the same language, the same treatment of data, and the same revision methodology, but they differ for the representation of the knowledge bases involved.

Since the sensors are reliable and the world is locally consistent (no sensor can read p and $\neg p$ at once) we obtain that the K_{ij}s modalities are normal, which means they satisfy the following axiom

$$\mathrm{K}_{ij}(A \rightarrow B) \rightarrow (\mathrm{K}_{ij}A \rightarrow \mathrm{K}_{ij}B)\ . \tag{1}$$

The axioms connecting K_{ij} with $\Box$ and $\Diamond$ are:

$$\left(\bigwedge_{i=1,j=i_1}^{i=n,j=i_m} \mathrm{K}_{ij}A\right) \to \Box A \qquad\qquad \left(\bigvee_{i=1,j=i_1}^{i=n,j=i_m} \mathrm{K}_{ij}A\right) \to \Diamond A \tag{2}$$

Each agent has at least a sensor able to perform measurements, and the supervisor supervises at least an agent, therefore from the above axioms we obtain $\Box A \to \Diamond A$. Due to the epistemic interpretation of the modal operators $\Box$ and $\Diamond$, it is natural to assume the common axioms for positive introspection ($\Box A \to \Box\Box A$), and negative introspection ($\Diamond A \to \Box\Diamond A$). It is possible that agents or the supervisor are equipped with on-board sensors; in this case we add the axiom $\Box A \to A$, so the resulting modal logics, in which we express the knowledge bases of the agents and the supervisor, are the well known systems $D45$ and $S5$.

In the SA-model we have to deal only with the knowledge base of the agent consisting of the pair $\mathcal{B}_a = \langle \mathcal{F}, \mathcal{R} \rangle$ where $\mathcal{F}$ is the set of facts collected by the sensors, and $\mathcal{R}$ is the set of rules. In the SAS-model we have a knowledge base for the supervisor, and a knowledge base for each agent. The knowledge base $\mathcal{B}_S$ of the supervisor is $\mathcal{B}_S = \langle \mathcal{F}, \mathcal{G}, \mathcal{E}_{a_1}, \dots, \mathcal{E}_{a_n} \rangle$ where $\mathcal{F}$ is the set of facts collected by the sensors; $\mathcal{G}$ is the set of global rules; and each $\mathcal{E}_{a_i}$ is a subset of $\mathcal{G}$ containing the global rules that do not hold for the agent a_i. The knowledge base $\mathcal{B}_{a_i}$ of an agent a_i is described in terms of the triple: $\mathcal{B}_{a_i} = \langle \mathcal{F}_{a_i}, \mathcal{L}_{a_i}, \mathcal{G}_{a_i} \rangle$ where $\mathcal{F}_{a_i}$ is the set of facts collected by the sensors; $\mathcal{L}_{a_i}$ is the set of agent's local internal rules; and $\mathcal{G}_{a_i} = \mathcal{G} - \mathcal{E}_{a_i}$ is the set of down-loaded rules, i.e., the rules passed to the agent by the supervisor.

We still have to see how data are passed to the agents and the supervisor; we identify a sensor with the set of information it has collected. So, if $p \in s_{ij}$ then $\mathrm{K}_{ij}p \in \mathcal{F}_{a_i}$, and then $\mathcal{F} = \bigcup_i \mathcal{F}_{a_i}$.

It is worth noting that all the pieces of information in the set of the data gathered by an agent or the supervisor are propositional and are transformed into modal form once passed to the supervisor. We argue that data in conditional and disjunctive form (e.g., $p \to q$, $p \vee q$) are meaningless while we accept negative data, for example $\neg p$. Let us suppose that our sensors are cameras, whose optical field consists of n pixels. We have a wall where we have drawn lines of various lengths. We move our camera in front of each line, then p stands for "the length of the line is n", where n is a given number of pixels. $\neg p$ means that the length of the line actually in front of the camera is not of n pixels. In general a negative data corresponds to a measurement beyond given bounds. Since we accept only measurements in conjunctive and negative form we can reduce the elements of each $\mathcal{F}$ into literals.

4 Revision of Finite Modal Knowledge Bases

We saw in section 2 that the revision mechanism is central to the behavior of the system. Different models have been described, depending on: a) which data are defeasible and which are assumed as certain; b) which modal characterization is given to the underlying theory.

The revision engine that will be described in this paragraph can perform modal revisions according to the constraints of finiteness and implementing different schemes of defeasibility. It is a computationally oriented procedure for revision of finite knowledge bases.

The classical AGM model ([1], [13]) has been recently criticized (see [14], [9], [7]) for being too liberal about constraints of finiteness and computation. The axiom of recovery, saying that the result of adding a piece of knowledge to the result of contracting it from a knowledge base returns the original base, has been criticized as being unnecessarily demanding (see [15], [16] for a thorough analysis).

Our procedure is computationally oriented and does not satisfy the recovery axiom and yields a minimal change in the sense that the original rules of the knowledge base are retained, albeit in a modified form, as long as possible. The procedure as described here does not use information about entrenchment. It can yield non-trivial results also when no entrenchment information is available, contrary to what happens in the classical AGM approach, and it can be easily extended to manage them, when available.

We use revision as the primitive operator, although it is often argued that contraction should be used as a primitive operator (see [8]; see also [4] for a different view); however, it seems rather unnatural to suppose that our agents should change their minds about properties of the world were it not for the necessity of incorporating in their knowledge base a new fact. Exploration by means of sensors always yields new data to be added in some way into the existing corpus; it never offers a negative view. An exception should be made for the case in which a sensor tells its agent that a measurement already done is unreliable. But this amounts to going back to the rule set existing before that measurement, and revise it using all subsequent measurement except the one that was declared unreliable. This is not, however, the primitive case, but a rather complex and sophisticated one, which deserves the role of a derived operation.

There is one more reason for not choosing contraction as a primitive operation, and it is connected with the choice of dealing with modal revision. The standard account for contraction goes as follows. Let us assume that base B implies proposition α. Then, for some reason, α has to be abandoned. So, we have to contract α from B. The reason for relinquishing α, however, is not that $\neg\alpha$ is found to hold, otherwise we should revise B by $\neg\alpha$. Rather, we feel unsure as to which of α and $\neg\alpha$ should be maintained. For instance, we perform repeated measurements, and sometimes we get α and sometimes $\neg\alpha$. As we feel that both α and $\neg\alpha$ might be the case, we have no choice other than to contract α. This fits well with a modal approach, in which the only primitive operation is revision. Indeed, the revision/contraction contrast may be described in terms of modalities. The situation described can be restated in the following terms: the base B implies $\Box\alpha$; then, we perform repeated measurements, and sometimes we get α and sometimes $\neg\alpha$; this means that our theory must be *revised* by $\Diamond\alpha \wedge \Diamond\neg\alpha$, properly expressing the fact that we feel that both α and $\neg\alpha$ might be the case.

5 The Revision Procedure

As noted before, the revision procedure starts when a formula is added to a set of formulae, and some contradiction would arise if no modification is made.

Formulae are divided into two classes: defeasible and non-defeasible. For simplicity we shall call them, respectively, *rules* and *facts*, even if a non-defeasible formula might be, in fact, a rule (e.g., a down-loaded rule), and a fact could be defeasible (e.g., if we adopt the *update* point of view, in which only the last fact is non-defeasible). How this partition is made depends on the model of exchange and communication between the levels of the hierarchy. Both rules and facts are expressed in a modal propositional language. We set no restriction on the form of rules and facts. For the sake of simplicity, in the examples we shall assume that rules have the form $A \rightarrow B$, where A and B are arbitrary modal formulae. If this is not the case, a simple manipulation will do (for instance, replacing an arbitrary formula C with $\top \rightarrow C$, where $\top$ is the atomic constant for $true$).

The procedure is based on the following steps:

1. find all minimal inconsistent subsets of sub-formulae of the original set of formulae;
2. for any subset, weaken all the rules modally and propositionally;
3. reconstruct the set of formulae starting from the revised subsets.

The procedure yields a finite set; the process of finding the inconsistent subsets relies on the specific modal logic, and is computationally feasible thanks to the properties KEM, the method we shall use to determine the minimal inconsistent sets (see section 5.1).

The first step is rather standard in principle. Using KEM , however, makes it possible to construct all minimal inconsistent subsets of formulae at the same time the inconsistency is proved, resulting in a much better efficiency.

The second step, on the contrary, is not so common. When revision of finite bases is performed, it is rather standard to restore consistency by deleting one or more formulae from each inconsistent subset (that is, contracting the base) and then adding the new fact. Of course, deleting all of them results in a far from minimal mutilation of the knowledge base; on the other side, if no extra-logical information is supplied as to which formula has to be deleted, deleting all of them is the only alternative to non-determinism. The choice is between a revision based on the so-called safe contraction (too demanding) and a non-deterministic revision. Our proposed algorithm keeps the flavour of a safe revision, in that it operates on all the formulae in the subsets, but does not delete them: we simply modify them in order to recover consistency. We want to avoid non-determinism, which might result in a disaster for the agents, at the same time retaining as much information is possible.

The third step is, again, rather standard: minimal inconsistent sets are deleted and replaced by the modified sets.

The resulting formulae are still divided into rules and facts, so the base can be used as a base for further revisions. However, it should be noted that this division into facts and rules is not essential to the process. If a fact is the only new piece of code to be added, and all other pieces are on the same par, the process yields again a reasonable result, contrary to what happens in the AGM frame, where we end up with just the consequences of the new fact.

5.1 Modal Tableaux as Contradiction Finders

A well known family of theorem proving methods is based on tableaux. A tableau is a tree whose nodes are labelled by formulae related to the formula to be proved. Tableaux-based methods aim to prove a formula by showing that there are no counter-examples to it, i.e., by showing that any assignment of truth values to the variables makes the negation of the formula false. While tableaux are usually employed as theorem provers, they are, literally, contradiction finders. In fact, in the development of a tableau we try to show that the formula which labels the node under scrutiny cannot be satisfied by any assignment of truth values. If the root of the tree is labelled not by a formula but by a set of formulae, showing that all subtrees are closed we prove just the fact that the set is inconsistent. In this case, we may employ the information gathered in the process of finding the inconsistency to restore the consistency of the set of formulae through modification of some of the formulae involved.

We shall use KEM which offers considerable advantages in terms of performance and flexibility in adapting to different modal systems (see [2]). In order to use KEM in the process of identifying the sources of contradiction, we augment its formalism by adding a mechanism able to record which subset of sub-formulae of the original set of formulae is being used in the development of the branch, and so the subset involved in the contradiction which closes the branch.

We start from a brief description of KEM (for a detailed exposition see [2]). KEM is a labelled tableaux system based on a mixture of natural deduction and tableaux rules which uses labels to simulate the accessibility relation, and a unification algorithm to determine whether two labels denote the same world. It can be also used to check the consistency of a set of formulae, and information extracted from the tree helps the solving of not immediate contradictions; elsewhere [3] a preferences strategy connected to KEM has been adopted for the same problem. KEM uses two kinds of atomic labels: a set of *constant world symbols*, $\Phi_C = \{w_1, w_2, \dots\}$ and a set of *variable world symbols*, $\Phi_V = \{W_1, W_2, \dots\}$ that might be combined into *path* labels. A path is a label with the following form (i, i'), where i is an atomic label and i' is either a path or a constant. Given a label $i = (k, k')$ we shall use $h(i) = k$ to denote the head of i, and $b(i) = k'$ to denote the body of i, where such notions are possibly applied recursively. $l(i)$, and $s^n(i)$ denote respectively the length of i (the number of world symbols it contains), and the sub-label (segment) of length n counting from right to left. As an intuitive explanation, we may think of a label $i \in \Phi_C$ as denoting a world (a *given* one), and a label $i \in \Phi_V$ as denoting a set of worlds (*any* world) in some Kripke model. A label $i = (k', k)$ may be viewed as representing a path from k to a (set of) world(s) k' accessible from k (or, equivalently, the world(s) denoted by k).

Labels are manipulated in a way closely related to the accessibility relation of the logic we are concerned with. To this end it is possible to define logic dependent label unifications σ_L, which will be used in the course of KEM proofs. We start by providing a substitution $\rho : \Im \mapsto \Im$ thus defined:

$$\rho(i) = \begin{cases} i & i \in \Phi_C \\ j \in \Im & i \in \Phi_V \\ (\rho(h(i)), \rho(b(i))) & l(i) > 1 \end{cases}$$

From ρ we define the unification σ from which it is possible to define the appropriate unifications for a wide range of modal and epistemic logics (see [2]), as follows:

$$\forall i, j, k \in \Im, (i, j)\sigma = k \text{ iff } \exists \rho : \rho(i) = \rho(j) \text{ and } \rho(i) = k$$

However, in this paper, we present only the σ_L-unifications for the logics we are concerned with, namely $L = D45, S5$.

$$(i,k)\sigma_{D45} = \begin{cases} ((h(i),h(k))\sigma,(s^1(i),s^1(k))\sigma) & l(i),l(k) > 1, \\ (i,k)\sigma & l(i),l(k) = 1 \end{cases} \tag{3}$$

$$(i,k)\sigma_{S5} = \begin{cases} ((h(i),h(k))\sigma,(s^1(i),s^1(k))\sigma) & (h(i),h(k))\sigma \neq (s^1(i),s^1(k)) \\ (h(i),h(k))\sigma & \text{otherwise} \end{cases} \tag{4}$$

Example 1. It can be seen that $((W_1,w_1),(W_2,w_1))\sigma_{D45}$, but the paths (W_1,w_1) and w_1 do not σ_{D45}-unify; this corresponds to the fact that $\Box A \to \Diamond A$ holds in $D45$, but $\Box A \to A$ does not.

In defining the inference rules of KEM we shall use *labelled signed formulae*, where a labelled signed formula (LS-formula) is an expression of the form X, i where X is a signed formula and i is a world label. Given a modal formula A, i, the LS-formulae TA, i and FA, i represent the assertion that A is true or, respectively, false at the world(s) denoted by i. TA and FA are conjugate formulae. Given a signed formula X, by X^C we mean the conjugate of X.

In the following table signed formulae are classified according to Smullyan-Fitting unifying notation [6].

α	α_1	α_2	β	β_1	β_2
$TA \wedge B$	TA	TB	$FA \wedge B$	FA	FB
$FA \vee B$	FA	FB	$TA \vee B$	TA	TB
$FA \to B$	TA	FB	$TA \to B$	FA	TB

ν	ν_0	π	π_0
$T\Box A$	TA	$T\Diamond A$	TA
$F\Diamond A$	FA	$F\Box A$	FA

We shall write $[\alpha_1, \alpha_2]$ and $[\beta_1, \beta_2]$ to denote the two components of a α-formula (respectively, of a β-formula).

KEM builds a tree whose root is labelled with the set of signed formulae $\{TA_1, \ldots, TA_n\}$ and the label w_1, corresponding to the assertion that all propositions $A_1, \ldots, A_n$ in the original rule set are true in the actual world. Branches are built by means of inference rules which derive new signed formulae which hold in specific sets of worlds. In doing so, they build new signed formulae and new world label strings. While the rules for deriving new signed formulae depend only on propositional logic, the rules for deriving world label strings depend on the specific modal logic at hand. A branch is closed when it contains a signed formula X and its conjugate X^C which hold in the same world. If all branches are closed, the tree is closed, and the root is contradictory.

A characteristic of KEM is the analyticity property, that is, all signed formulae are sub-formulae of one of the original formulae. This is accomplished by limiting the cut-rule to one component of a β formula.

In order to gather information about the inconsistencies of the set of rules, we enrich KEM with three new sets of labels. The first one records the (component of the) original signed formula from which the signed formula labelling the node derives. The second one records the set of (components of the) original signed formulae used in deriving the signed formula labelling the node. The third one records the *ad hoc* assumptions made by applications of the cut rule. These three labels will be denoted by l, sl and c, respectively.

A difference from the usual procedure of KEM is that even if all the branches are closed, we continue deriving new nodes until all the original formulae are used in all branches. When the procedure stops, we have a number of nodes where the branches close. The sl label, together with the c label, identifies a minimal inconsistent set of rules. The rules in sl are to be revised in order to restore consistency.

$$\alpha\text{-rule} \qquad \begin{array}{ccccc} [\alpha_1,\alpha_2] & i & l & sl & c \\ \hline \alpha_1 & i & l.\alpha_1 & sl-\{l\}\cup\{l.\alpha_1\} & c \\ \alpha_2 & i & l.\alpha_2 & sl-\{l\}\cup\{l.\alpha_2\} & c \end{array}$$

$$\beta\text{-rule} \qquad \begin{array}{ccccc} [\beta_1,\beta_2] & i & l & sl & c \\ \beta_n^C(n=1,2) & j & l' & sl' & c' \\ \hline \beta_{3-n} & (i,j)\sigma_L & l.\beta_{3-n} & sl\cup sl'-\{l\}\cup\{l.\beta_n\} & c\cup c' \end{array}$$

$$\nu\text{-rule} \qquad \begin{array}{ccccc} \nu & i & l & sl & c \\ \hline \nu_0 & (i',i) & l & sl & c \end{array}\; i'\in\Phi_V \text{ new}$$

$$\pi\text{-rule} \qquad \begin{array}{ccccc} \pi & i & l & sl & c \\ \hline \pi_0 & (i',i) & l & sl & c \end{array}\; i'\in\Phi_C \text{ new}$$

$$\perp\text{-rule} \qquad \begin{array}{ccccc} X & i & l & sl & c \\ X^C & j & l' & sl' & c' \\ \hline \perp & (i,j)\sigma_L & nil & (sl\cup\{l\}-c)\cup(sl'\cup\{l'\}-c') & c\cup c' \end{array}$$

$$\text{cut-rule} \qquad \begin{array}{ccccccccccc} & & & & & [\beta_1,\beta_2] \quad i \quad l \quad sl \quad c & & & & & \\ \hline \beta_1 & i & l.\beta_1 & \emptyset & c\cup\{l.\beta_1\} & & \beta_1^C & i & l.\beta_1 & \emptyset & c\cup\{l.\beta_1\} \end{array}$$

(similarly when β_2 is used)

Soundness and completeness for the above calculus are given in [2].

The minimal contradictory sets may contain labels of the form $l.s$ where l is the number of a formula and s is a string built by tokens belonging to the set $\{\alpha_1,\alpha_2,\beta_1,\beta_2\}$, with dots between. This means that the subformula of l identified by s is responsible for the contradiction. The other components of the formula l can be retained; only $l.s$ has to be weakened. So the structure of the labels that tells us which components have to be weakened.

5.2 How To Weaken the Formulae Responsible for Contradiction

Given a set of rules and facts which yields a contradiction, we may restore consistency by weakening the rules, that is, the defeasible ones. Rules may be weakened both modally and propositionally. Some of the elements of a minimal contradictory set might be facts, that is, non-defeasible formulae. If all elements are facts, no revision can be made; there is no way of restoring consistency. If, on the contrary, at least some of the formulae are rules, we can weaken them and restore consistency.

Let $\{A_{ki}\to B_{ki}\}$ be the set of rules in the k-th minimal contradictory set, and $\{F_j\}$ be the set of facts in the k-th minimal contradictory set. The propositional weakening of $A_{ki}\to B_{ki}$ is the rule $(\bigvee_j \neg F_j \wedge A_k)\to B_{ki}$. As all facts F_j hold, the antecedent of the weakened rule is false, and the inference of B_{ki} is blocked. The exact form depends on the sub-formulae involved. In turn, the exact form of the modal weakening of $\alpha_i \to \beta_i$ depends on the modal system we are working in.

In general, the modal weakening of $A_i \to B_i$ includes an antecedent weakening and a consequent weakening, which are, in a sense, dual of each other. In order to fix ideas, let's consider the antecedent weakening. We want to substitute $A_{ki} \to B_{ki}$ with $\sigma_\downarrow(A_{ki}) \to B_{ki}$, where $\sigma_\downarrow(A_{ki})$ is a modal expression such that $\sigma_\downarrow(A_{ki}) \to A_{ki}$ but the converse does not hold; $\sigma_\downarrow(A_{ki}) \to B_{ki}$ is weaker than $A_{ki} \to B_{ki}$, in the sense that it is more difficult to find a counter-example to $\sigma_\downarrow(A_{ki}) \to B_{ki}$ than it is for $A_{ki} \to B_{ki}$; this may be expressed by saying that the set of models for $\sigma_\downarrow(A_{ki}) \to B_{ki}$ is greater than the set of models for $A_{ki} \to B_{ki}$. However, if we want to perform a revision according with some criteria of minimality, we need to put some constraints on the choice of $\sigma_\downarrow$. It seems to us that the following constraints are reasonable:

a) $\sigma_\downarrow(A_{ki})$ is a modal function of A_{ki}, that is, can be built without any information other than A_{ki}
b) $\sigma_\downarrow(A_{ki})$ is a positive modal function, that is, no negation except those possibly contained in A_{ki} should be used.
c) We should use the weakest modal expression among those satisfying a) and b), in order to obtain some kind of minimal revision

There are two reasons for using only positive modal functions. First, in using arbitrary modal functions we might turn a true formula into a false one. Assume that we revise $A \to B$ by $A \wedge \Diamond\neg A \to B$. Now, if A happens to be an identically true formula, it is true also in all the accessible worlds, and $A \wedge \Diamond\neg A$ is identically false. While the original rule was equivalent to B, the new one is simply irrelevant. Second, by limiting ourselves to positive modal functions we impose constraints on the set of worlds in which the antecedent must hold in order to derive the consequent, while using arbitrary functions we are no more able to give such an interpretation to our revised rule.

Condition c) may be difficult to satisfy, because it may be difficult to determine a unique minimal modal expression. We already said that we want to avoid non-determinism and excessive mutilation of the knowledge base. Using these guidelines, we satisfy condition c) by means of the following construction (given a modal logic):

c.1) build the poset $[D(A),\to]$, where $D(A)$ is the set of the positive modal functions of A;
c.2) intersect all the chains (linear ordered subsets) of $[D(A),\to]$ that include A itself; this will be a chain itself;
c.3) $\sigma_\downarrow(A)$ is the greatest element of the chain such that $\sigma_\downarrow(A) \to A$; it is the element nearest to A "from below".

Whether step c.3) can be performed or not depends, of course, on the system of logic we work in. However, it can be shown that $\sigma_\downarrow(A)$ exists for some common logics. For example for the logics $D45$ and $S5$ described in section 3, it possible to define exactly the result of the weakening process. This can be expressed using a chain of modal functions of an arbitrary formula ϕ. It can be proved that the chains of modal functions in $D45$ and $S5$ are

$\phi \wedge \Box\phi \to \phi \wedge \Diamond\phi \to \phi \to \phi \vee \Box\phi \to \phi \vee \Diamond\phi$	$\Box\phi \to \phi \to \Diamond\phi$
Chain of modal functions in $D45$	Chain of modal functions in $S5$

Steps c.2) and c.3), respectively, guarantee safeness (intersecting all the chains is similar to weakening all the formulae) and minimality.

Similarly the weakening of the consequent results by substituting B_{ki} with $\sigma_{\uparrow}(B_{ki})$, a modal expression built using only B_{ki} and no extra negations such that $B_{ki} \rightarrow \sigma_{\uparrow}(B_{ki})$ but the converse does not hold.

As a simple example, let us suppose that the rule: $A \rightarrow B$ is responsible for a contradiction; let us also assume that the only distinct modal affirmative functions of x in the language are $\Box x$ and $\Diamond x$, and that there is a predicate C whose value is *true*. Then, the original rule can be weakened by substituting it with three rules:

$$\Box A \rightarrow B \qquad\qquad A \rightarrow \Diamond B \qquad\qquad A \wedge \neg C \rightarrow B$$

Even if the three rules allow a number of non-trivial derivations, they no longer allow the derivation of B.

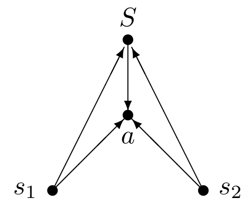
Some justification is due for using modal expressions with no extra negations. This amounts to using positive modal functions as weakening sensors. There are two reasons for that. In revising the antecedent, we use the weakest suitable modal expression (the strongest for revising the consequent). This might be insufficient to block the inference responsible for the contradiction. If an inconsistency is still found, we must use a "stronger" weakening, if available. If not, we have no more options, other than to discard the rule. In $S5$ we have no second options, due to the fact that $\Box\Box\alpha = \Box\alpha$. In $D45$, however, more possibilities are at hand. If only one of the components of a β-formula is included in the contradictory set, then the other can be safely weakened the standard way, while the component actually implied could need a stricter weakening.

It must be noted that we add *facts*, but only *rules* are revised. This means that if an inconsistent set includes only facts, no revision is possible: inconsistent facts cannot be reconciled.

5.3 How to Reconstruct the Set of Formulae Starting from the Revised Subsets

In order to reconstruct a new set of rules from the revised rules in the contradictory sets, we have first of all to make sure to include all the sub-rules not to be revised. This can be easily done by adding to the original set the complement of the rules in one of the contradictory sets. Some examples will clarify the matter. Let $A_1 \vee A_2 \rightarrow B_1 \wedge B_2$ be the original rule l, and let $l.\beta_1.\alpha_2$ be in the contradictory set. This means that the sub-rule actually implied is $A_2 \rightarrow B_1 \wedge B_2$ We find it by following the structure of the label $l.\beta_1.\alpha_2$: break the rule in the first β-component, taking the second α-component. If the labels were $l.\beta_1.\alpha_2$ and $l.\beta_2.\alpha_1$ then the rule implied would be $A_2 \rightarrow B_1$. In our case, the other sub-rule $A_1 \rightarrow B_1 \wedge B_2$ may be safely added to the original set of rules. Then we have to add the revised rules. In our example, we should add $A_2 \wedge \bigvee_j \neg F_j \rightarrow B_1 \wedge B_2$ as the propositional weakening, and the rule $\Box A_2 \rightarrow B_1 \wedge B_2$ as the modal weakening. As the third step, we must delete from the set of rules all sub-rules in the contradictory sets and all the parent rules. The necessity of also deleting the sub-rules in the contradictory sets stems from the possibility of reintroducing a rule *piecewise*, one sub-rule from each of the sets. Then we have to check for consistency of the modified rules. It is enough to check the consistency of the rules resulting from the revision on the rules in the contradictory sets plus the ad-hoc assumptions (the set labelled by c in the tableaux).

6 Examples



Example 2 In this example, we employ the most simple but not trivial structure, i.e., a supervisor S, an agent a and two sensors s_1 and s_2 arranged as depicted beside, we show how the revision procedure works, assuming $S5$ as the logic for the agent and the supervisor.

In this example we deal with the knowledge bases, i.e., $\mathcal{B}_a = \langle \mathcal{F}_a, \mathcal{L}_a, \mathcal{G}_a \rangle$ for the agent a, and $\mathcal{B}_S = \langle \mathcal{F}, \mathcal{G}, \mathcal{E}_a \rangle$

We start with $\mathcal{L}_a = \{\Box p \to \Box q\}$, $\mathcal{G} = \{\Diamond q \to \Box r\}$, and $\mathcal{E}_a = \emptyset$, therefore $\mathcal{G}_a = \mathcal{G}$. The sensors gather the following information: $s_1 = \{p\}$ and $s_2 = \{p, \neg q\}$ which are passed to the agent in the following form: $\{\mathrm{K}_1 p, \mathrm{K}_2 p, \mathrm{K}_2 \neg q\}$ that means $\mathcal{F}_a = \{\Box p, \Diamond \neg q\}$. The agent checks the consistency of its own knowledge base by the means of the KEM-tree starting with the union of the elements, namely $\mathcal{L}_a \cup \mathcal{G}_a \cup \mathcal{F}_a$.

$\mathcal{G}_a = \{$	$1\ T\Diamond q \to \Box r$	w_1 1	1	$\emptyset$
$\mathcal{L}_a = \{$	$2\ T\Box p \to \Box q$	w_1 2	2	$\emptyset$
$\mathcal{F}_a = \{$	$3\ T\Box p$	w_1 3	3	$\emptyset$
	$4\ T\Diamond \neg q$	w_1 4	4	$\emptyset$
	$5\ T\Box q$	w_1 $2.\beta_2$	$2.\beta_1, 3$	$\emptyset$
	$6\ Fq$	(w_2, w_1) 4	4	$\emptyset$
	$7\ Tq$	(W_1, w_1) $2.\beta_2$	$2.\beta_1, 3$	$\emptyset$
	$8\ \bot$	(w_2, w_1) $-$	$2.\beta_1, 2.\beta_2, 3, 4$	$\emptyset$

Notice that we have deleted all the inessential steps. In fact, it is immediate to see that no other contradictions can be derived from the above tree. The set of formulae responsible for the contradiction is $\{2.\beta_1, 2.\beta_2, 3, 4\}$; however only $2.\beta_1$ and $2.\beta_2$ should be revised in so far as 3 and 4 are facts. We apply the revision function obtaining $\sigma_\uparrow(\Box p) \to \Box q$ and $\Box p \to \sigma_\downarrow(\Box q)$ The first fails, $\Box p$ is already at the top of the chain; whereas the second succeeds, being $q = \sigma_\downarrow(\Box q)$. Therefore the revised set of internal rules consists of $\mathcal{L}'_a = \{\Box p \to q\}$ At this point the agent has restored consistency and the sensors may collect new pieces of information. Let us assume that the new data are $s_1 = \{p, \neg r\}$ and $s_2 = \{p, \neg q\}$ The new set of facts turns out to be $\mathcal{F}'_a = \{\Box p, \Diamond \neg q, \Diamond \neg r\}$ Again, the agent runs the KEM tree for its knowledge base.

$\mathcal{G}_a = \{$	$1\ T\Diamond q \to \Box r$	w_1 1	1	$\emptyset$
$\mathcal{L}'_a = \{$	$2\ T\Box p \to q$	w_1 2	2	$\emptyset$
$\mathcal{F}'_a = \{$	$3\ T\Box p$	w_1 3	3	$\emptyset$
	$4\ F\Box q$	w_1 4	4	$\emptyset$
	$5\ F\Box r$	w_1 5	5	$\emptyset$
	$6\ Tq$	w_1 $2.\beta_2$	$2.\beta_1, 3$	$\emptyset$
	$7\ F\Diamond q$	w_1 $1.\beta_1$	$1.\beta_2, 5$	$\emptyset$
	$8\ Fq$	(W_1, w_1) $1.\beta_1$	$1.\beta_2, 5$	$\emptyset$
	$9\ \bot$	w_1 $-$	$1.\beta_{1,2}, 2\beta_{1,2}, 3, 5$	$\emptyset$

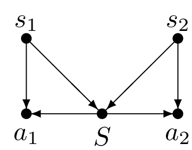
The contradiction arises from $\{1.\beta_1, 1.\beta_2, 2.\beta_1, 2.\beta_2, 3, 5\}$, but only $2.\beta_1, 2.\beta_2$ have to be revised by the agent: 3 and 5 are facts and 1 is a global rule that can be revised only by the supervisor. The revision function leads to $\sigma_\uparrow(\Box p) \to q$, $\Box p \to \sigma_\downarrow(q)$, and $\Box p \wedge \Box r \to q$ The first is not applicable for the same reason of the previous case, the second produces $\Box p \to \Diamond q$ and the third is the propositional weakening of 2. The resulting state

$$\mathcal{F}'_a = \{\Box p, \Diamond\neg q, \Diamond\neg r\} \quad \mathcal{L}''_a = \{\Box p \to \Diamond q, \Box p \wedge \Box r \to q\} \quad \mathcal{G}_a = \{\Diamond q \to \Box r\}$$

is inconsistent.

$\mathcal{G}_a =$	$1\ T\Diamond q \to \Box r$	$w_1\ 1$	1	$\emptyset$
$\mathcal{L}''_a =$	$2\ T\Box p \to \Diamond q$	$w_1\ 2$	2	$\emptyset$
	$3\ T\Box p \wedge \Box r \to q$	$w_1\ 3$	3	$\emptyset$
$\mathcal{F}'_a =$	$4\ T\Box p$	$w_1\ 4$	4	$\emptyset$
	$5\ F\Box q$	$w_1\ 5$	5	$\emptyset$
	$6\ F\Box r$	$w_1\ 6$	6	$\emptyset$
	$7\ T\Diamond q$	$w_1\ 2.\beta_2$	$2.\beta_1, 4$	$\emptyset$
	$8\ T\Box r$	$w_1\ 1.\beta_1$	$1.\beta_2, 2.\beta_1, 4$	$\emptyset$
	$9\ \bot$	$w_1\ -$	$1.\beta_{1,2}, 2.\beta_1, 4, 6$	$\emptyset$

It is easy to see that the inconsistency arises from the conjunction of local and global rules, namely from $\Box p \to \Diamond q$ and $\Diamond q \to \Box r$, therefore the agent notifies the inconsistency to the supervisor. However the supervisor recognizes a consistent state, being $\mathcal{F} = \mathcal{F}_a$ and $\mathcal{G} = \mathcal{G}_a$. At this point the agent has to revise again the culprit rule (i.e., $\Box p \to \Diamond q$); unfortunately the only way to revise it is to delete it, no more modal weakenings are possible. The resulting set of local rules $\mathcal{L}'''_a$ consists of $\Box p \wedge \Box r \to q$.



Example 3 In this example we assume a slightly more complex structure to illustrate the use of exceptions. In this framework we have a supervisor S, two agents a_1 and a_2, and each agents has a single sensor. According to our model the knowledge bases are: $\mathcal{B}_S = \langle \mathcal{F}, \mathcal{G}, \mathcal{E}_{a_1}\mathcal{E}_{a_2}\rangle$, $\mathcal{B}_{a_1} = \langle \mathcal{F}_{a_1}, \mathcal{L}_{a_1}, \mathcal{G}_{a_1}\rangle$, and $\mathcal{B}_{a_2} = \langle \mathcal{F}_{a_2}, \mathcal{L}_{a_2}, \mathcal{G}_{a_2}\rangle$ where $\mathcal{L}_{a_1} = \mathcal{L}_{a_2} = \emptyset$, $\mathcal{E}_{a_1} = \mathcal{E}_{a_2} = \emptyset$, and $\mathcal{G} = \{\Box p \to \Diamond q\}$. Therefore $\mathcal{G} = \mathcal{G}_{a_1} = \mathcal{G}_{a_2}$. The sensors collect the following data: $s_1 = \{p, \neg q\}$ $s_2 = \{p, q\}$, then the sets of local facts are: $\mathcal{F}_{a_1} = \{\Box p, \Box\neg q\}$ and $\mathcal{F}_{a_2} = \{\Box p, \Box q\}$ It is immediate to see that $\mathcal{F}_{a_1}$ and $\mathcal{G}_{a_1}$ are inconsistent, and the contradiction is due to the global rule $\Box p \to \Diamond q$. The supervisor gathers the data from all the sensors obtaining $\{\mathrm{K}_1 p, \mathrm{K}_2 p, \mathrm{K}_1\neg q, \mathrm{K}_2 q\}$ which implies $\mathcal{F} = \{\Box p, \Diamond\neg q, \Diamond q\}$. However, $\mathcal{F} \cup \mathcal{G}$ is consistent, so the supervisor adds $\Box p \to \Diamond q$ to the exceptions of a_1 ($\mathcal{E}_{a_1}$). The new knowledge bases are: $\mathcal{B}_S = \langle \mathcal{F}, \mathcal{G}, \mathcal{E}'_{a_1}, \mathcal{E}_{a_2}\rangle$ and $\mathcal{B}_{a_1} = \langle \mathcal{F}_{a_1}, \mathcal{L}_{a_1}, \mathcal{G}'_{a_1}\rangle$, where $\mathcal{E}_{a_1} = \{\Box p \to \Diamond q\}$, and, consequently $\mathcal{G}'_{a_1} = \mathcal{G} - \mathcal{E}_{a_1} = \emptyset$.

The procedure for dealing with exceptions can be viewed as a special kind of modal weakening. Due to the equivalence $(\top \to A) \equiv A$ global rules can be conceived of as consequents of conditional rules whose antecedent is an always true formula. We then apply the revision function $\sigma_\downarrow(A)$ obtaining $\Diamond A$, which means that A holds somewhere. $\Diamond A$ restores consistency, but is too weak for our purposes, we do not know where A holds. However the supervisor knows the agent where the exception does not hold, so instead of changing the formula, it adds it to the set of exceptions.

7 Conclusion

In this paper a model of theory revision based on a hierarchy of agents is explored. In order to coordinate data acquisition from different agents, a modal language is used. We extend to the modal case ideas originally put forth for dealing with purely propositional knowledge. The problem of combining data from different sources is important per se, and a long standing priority for AI. Besides, the hierarchical models may be seen as first steps toward a fully distributed model, in which each agent builds and maintains a model of the other agents' knowledge.

Acknoweledgements

This research was partially supported by the Australia Research Council under Large Grant No. A49803544.

References

1. Alchourrón, C.E., Gärdenfors, P., and Makinson, M. (1985): On the logic of theory change: Partial meet contraction and revision functions, *The Journal of Symbolic Logic, 50*: 510–530.
2. Artosi, A., Benassi, P., Governatori, G., and Rotolo, A. (1996): Labelled Proofs for Quantified Modal Logic. In J.J. Alferes, L. M. Pereira, and E. Orlowska (eds.) *Logics in Artificial Intelligence*, Springer-Verlag, Berlin: 70–86.
3. Artosi, A., Governatori, G., and Sartor, G., (1996): Towards a Computational Treatment of Deontic Defeasibility. In M. Brown and J. Carmo (eds). *Deontic Logic, Agency and Normative Systems*. Springer-Verlag, Berlin: 27–46.
4. Boutilier, C.: (1996) Iterated Revision and Minimal Change of Conditional Beliefs, *Journal of Philosophical Logic, 25*: 263–305.
5. Chellas, B.F. (1980): *Modal Logic: An Introduction*, Cambridge University Press
6. Fitting, M., (1983): *Proof Methods for Modal and Intuitionistic Logics*. Reidel, Dordrecht.
7. Fuhrmann, A.(1991): Theory Contraction through Base Revision, *Journal of Philosophical Logic, 20*: 175–203.
8. Fuhrmann, A. and Hansson, S.O. (1994): A Survey of Multiple Contractions, *The Journal of Logic, Language and Information 3*: 39–75.
9. Hansson, S.O. (1992a): In defense of Base Contraction, *Synthese, 91*: 239–245.
10. Hughes, G. and Cresswell, M. (1996). *A New Introduction to Modal Logic*, Routledge, London.
11. Katsuno, H. and Mendelzon, A.O. (1992): On the Difference between updating a Knowledge Base and revising it, in Gärdenfors, P. (ed.), *Belief Revision*, Cambridge University Press.
12. Van Linder, B., Van der Hoek, W. and Meyer, J.-J. C. (1997): Seeing is Believing. And So Are Hearing and Jumping, *Journal of Logic, Language and Information, 6*: 33–61.
13. Makinson, D. (1989): General theory of cumulative inference, in Reinfrank, M. et al. (eds), *Non-Monotonic Reasoning — Proceedings of the 2nd International Workshop*, Springer.
14. Nebel, B. (1992): Syntax Based Approaches to Belief Revision, in Gärdenfors, P. (ed.), *Belief Revision*, Cambridge University Press.
15. Tennant, N.(1994): Changing the Theory of Theory Change: Towards a Computational Approach, *British Journal for the Philosophy of Science, 45*: 865–897.
16. Tennant, N. (1997): On Having Bad Contractions, or: No Room for Recovery, *Journal of Applied Non-Classical Logics, 7*: 241–265.