Taming Complex Beliefs*

Barbara Dunin-Kęplicz[1,2] and Andrzej Szałas[1,3]

[1] Institute of Informatics, University of Warsaw
Banacha 2, 02-097 Warsaw, Poland
{keplicz,andrzej.szalas}@mimuw.edu.pl
[2] Institute of Computer Science
Polish Academy of Sciences, Warsaw, Poland
[3] Department of Computer and Information Science, Linköping University
SE-581 83 Linköping, Sweden

Abstract. A novel formalization of beliefs in multiagent systems has recently been proposed by Dunin-Kęplicz and Szałas. The aim has been to bridge the gap between idealized logical approaches to modeling beliefs and their actual implementations. Therefore the stages of belief acquisition, intermediate reasoning and final belief formation have been isolated and analyzed. In conclusion, a novel semantics reflecting those stages has been provided. This semantics is based on the new concept of epistemic profile, reflecting agent's reasoning capabilities in a dynamic and unpredictable environment. The presented approach appears suitable for building complex belief structures in the context of incomplete and/or inconsistent information. One of original ideas is that of epistemic profiles serving as a tool for transforming preliminary beliefs into final ones. As epistemic profile can be devised both on an individual and a group level in analogical manner, a uniform treatment of single agent and group beliefs has been achieved.

In the current paper these concepts are further elaborated. Importantly, we indicate an implementation framework ensuring tractability of reasoning about beliefs, propose the underlying methodology and illustrate it on an example.

1 Beliefs in Multiagent Systems

During the past years awareness has been intensively investigated both from the theoretical as well as from the practical perspective. Its importance manifested itself especially in the context of cooperating teams of agents or other mixed groups in the context of intelligent, autonomous systems. In multiagent systems, agents' awareness is typically expressed in terms of different (combinations of) beliefs about

- the environment;
- an agent itself;
- other agents/groups involved.

Such beliefs are built using various forms of observations, communication and reasoning [2,12,13,15,36]. Existing modern, fine-grained logic-based approaches typically

* Supported by the Polish National Science Centre grants 2011/01/B/ST6/02769 and 6505/B/T02/2011/40.

N.T. Nguyen (Ed.): Transactions on CCI XI, LNCS 8065, pp. 1–21, 2013.

exploit rather subtle (combinations of) multi-modal logics [16,18,19,20]. Unfortunately this usually leads to high complexity of reasoning that is unacceptable from the point of view of their implementation and use. In fact, the underlying semantical structures are rather abstract and hardly reflect the way beliefs are acquired and finally formed. To make it even worse, in many applications one needs to take into account relevant features of perception, including:

– limited accuracy of sensors and other devices;
– restrictions on time and other resources affecting measurements;
– unfortunate combinations and unpredictability of environmental conditions;
– noise, limited reliability and failure of physical devices.

In multiagent systems during belief formation initial and intermediate beliefs are confronted with other beliefs originating from a variety of sources. The resulting beliefs can then substantially deviate from the initial ones. Moreover, there might still exist areas of agents' ignorance and inconsistencies. A low quality of information does not waive agents' responsibility of decision making. Therefore, reducing the areas of ignorance and inconsistencies is vital. In modern systems this can be accomplished in many different ways, including

– a variety of reasoning methods;
– belief exchange by communication;
– belief fusion;
– supplementary observations.

Apparently there is no guarantee to acquire the whole necessary information and/or to resolve all inconsistencies. Information may still remain partly unknown and/or inconsistent. Such situations may be sorted out by the use of

– paraconsistent models allowing for inconsistencies and lack of information;
– nonmonotonic reasoning techniques for completing missing information and resolving inconsistencies.

However, both paraconsistent and nonmonotonic reasoning, in their full generality, are intractable [5,17,21,25]. This naturally restricts their use in multiagent systems and calls for a shift in perspective. In [10] we proposed a novel framework for flexibly modeling beliefs of heterogenous agents, inspired by knowledge representation and deductive database techniques.

The key abstraction is that of *epistemic profiles* reflecting agent's individual reasoning capabilities. In short, epistemic profile defines a schema in which an agent reasons, deals with conflicting information and deals with its ignorance. These skills are achievable by combining various forms of reasoning, including belief fusion, disambiguation of conflicting beliefs or completion of lacking information. This rich repertoire of available methods enables for heterogeneity of agents' reasoning characteristics. More importantly, the same approach may be applied to groups of agents or even more complex mixed groups, allowing for uniform treatment of these, essentially different, cases.

In the current paper we show how the framework of [10] can serve as a basis for actual implementations and ensure tractability of reasoning. This is achieved by representing sets of preliminary and final beliefs as well as epistemic profiles using the deductive databases machinery of the 4QL query language [22,23,24,35].

The rest of this paper is structured as follows. In Section 2 our approach to structuring beliefs is outlined and motivated. Next, in Section 3, formal syntax and semantics of the basic language are defined. The pragmatics of its use in multiagent systems is discussed in Section 4. Section 5 is devoted to distributed beliefs. Implementation issues and complexity are addressed in Section 6. Finally, Section 7 concludes the paper.

The current paper is an extended and revised version of papers [10,11].

2 Structuring Beliefs

In the sequel, belief formation by agents will be unveiled. In the idealized logical approaches to agency, this paradigmatic part of agents' activity seems to be, at least partly, neglected. We analyze and model this process from the very beginning, that is from agents' perception and other kinds of basic beliefs.

The basis for the framework is formed by semantical structures reflecting the processes of an agent's belief acquisition and formation. Namely, an agent starts with *constituents*, i.e., sets of beliefs acquired by:

- perception;
- expert supplied knowledge;
- communication with other agents;
- other ways.

Next, the constituents are transformed into *consequents* according to the agent's *individual epistemic profile*.

While building a multiagent system, lifting beliefs to the group or even more complex level is substantial. As regards belief formation, it would be perfect to reach a conceptual compatibility between individual and group cases. Assuming that *the group epistemic profile* is set up, analogical individual and group procedures are then applicable for defining belief fusion methods, where:

- consequents of group members become constituents at the group level;
- such constituents are further transformed into group consequents.

Observe that this way various perspectives of agents involved are taken into consideration and merged. Moreover, we use the same underlying semantical structures for groups and individuals. The only requirement is that all epistemic profiles of complex structures are fixed. This way a uniform approach applies to groups of groups of agents or to mixed groups of individuals and other complex topologies.

Example 2.1. Consider an agent equipped with a sensor platform for detecting air pollution and two different sensors for measuring the noise level. The agent has also some information about the environment, including places in the neighborhood, etc. The task is to decide whether conditions in the tested position are healthy.

It is natural to consider, among others, three constituents:

- C_p gathering beliefs about air pollution at given places, in terms of $P(x, y)$ indicating the pollution level y at place x, where $y \in \{low, moderate, high\}$;
- C_n gathering beliefs about noise level at given places, in terms of $N_i(x, y)$ indicating the noise level y at place x, as measured by a sensor $i \in \{1, 2\}$, where $y \in \{low, moderate, high\}$;
- C_e gathering information about the environment in terms of $Cl(x, y)$ indicating that place x is close to a place characterized by y, where $y \in \{pollutive, noisy, neutral\}$.

For example, we may have:

$$C_p = \{P(a, low)\}, \quad C_n = \{N_1(a, high)\},$$
$$C_e = \{\neg Cl(a, noisy), Cl(a, neutral), Cl(a, pollutive)\}.$$

Note that we have no information from the second noise sensor (no literal $N_2()$ is given) and somehow inconsistent information as to the pollution level (C_p indicates low level, but according to C_e the agent is close to a pollutive location). Also there is an implicit disagreement between $N_1(a, high)$ appearing in C_e and $\neg Cl(a, noisy)$ appearing in C_e, which may be caused by a defective information source.

Based on constituents, the agent has to decide whether the situation is healthy or not (and include the thus obtained belief to the set of consequents). For example, the agent may accept

$$F = \{\neg S(a, healthy), S(a, healthy)\}$$

as its consequent, i.e., it may have inconsistent beliefs about the issue whether the situation at place a is healthy. $\lhd$

3 Syntax and Semantics

Inconsistency in common-sense reasoning attracted recently many logicians. To model inconsistencies, a commonly used logic is the four-valued logic proposed in [4]. However, as discussed, e.g., in [8,37], this approach is problematic in many applications. In particular, disjunction and conjunction deliver results which can be misleading for more classically oriented users.

On the other hand, our approach is strongly influenced by ideas underlying the 4QL query language [22,23,24] which does not share such problems. 4QL is a rule-based DATALOG$^{\neg\neg}$-like query language that provides simple, yet powerful constructs for expressing nonmonotonic rules reflecting, among others, default reasoning, autoepistemic reasoning, defeasible reasoning, local closed world assumption, etc. [22]. 4QL enjoys tractable query computation and captures all tractable queries. Therefore, 4QL is a natural implementation tool creating a space for a diversity of applications. To our knowledge a paraconsistent approach to beliefs has mainly been pursued in the context of belief revision [26,30]. However, these papers use formalisms substantially different from ours (like models based on criteria and rationality indexes [30] or relevant logic [26]).

Most of the approaches to modeling beliefs in logic start with variants of Kripke structures [12,15,16,18,19,34,36], where possible worlds are total and consistent. This

Table 1. Truth tables for $\wedge$, $\vee$, $\rightarrow$ and $\neg$ (see [37,22])

$\wedge$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$
$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{f}$
$\mathfrak{u}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$
$\mathfrak{i}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{i}$
$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$

$\vee$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$
$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$
$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$
$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{i}$	$\mathfrak{t}$
$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$

$\rightarrow$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{i}$	$\mathfrak{t}$
$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$
$\mathfrak{u}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{t}$
$\mathfrak{i}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{f}$
$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{t}$	$\mathfrak{t}$

	$\neg$
$\mathfrak{f}$	$\mathfrak{t}$
$\mathfrak{u}$	$\mathfrak{u}$
$\mathfrak{i}$	$\mathfrak{i}$
$\mathfrak{t}$	$\mathfrak{f}$

creates natural problems in modeling certain types of ignorance. For example, it is currently unknown whether, say, the Riemann's hypothesis is true. Therefore, to model this situation, we would have to create (at least) two possible worlds: one where the hypothesis is true and one where it is false. However, one of them would become inconsistent. In order to address such problems, modal frames involving non-standard worlds are considered [31,38]. However, our solution is simpler and leads to a substantial reduction of complexity.

In what follows all sets are finite except for sets of formulas.

We deal with the classical first-order language over a given vocabulary without function symbols. We assume that *Const* is a fixed set of constants, *Var* is a fixed set of variables and *Rel* is a fixed set of relation symbols.

Definition 3.1. *A* literal *is an expression of the form $R(\bar{\tau})$ or $\neg R(\bar{\tau})$, with τ being a sequence of arguments, $\bar{\tau} \in (Const \cup Var)^k$, where k is the arity of R.* Ground literals over *Const, denoted by $\mathcal{G}(Const)$, are literals without variables, with all constants in Const. If $\ell = \neg R(\bar{\tau})$ then $\neg \ell \stackrel{\text{def}}{=} R(\bar{\tau})$.* $\lhd$

Though we use the classical first-order syntax, the presented semantics substantially differs from the classical one. Namely,

- truth values $\mathfrak{t}, \mathfrak{i}, \mathfrak{u}, \mathfrak{f}$ (true, inconsistent, unknown, false) are explicitly present;[1]
- the semantics is based on sets of ground literals rather than on relational structures.

This allows one to deal with the lack of information as well as inconsistencies. As 4QL is based on the same principles, it can immediately be used as the implementation tool.

The semantics of propositional connectives is summarized in Table 1. Observe that definitions of $\wedge$ and $\vee$ reflect minimum and maximum w.r.t. the ordering:

$$\mathfrak{f} < \mathfrak{u} < \mathfrak{i} < \mathfrak{t}, \tag{1}$$

as advocated, e.g., in [7,22,37]. Such a truth ordering seems to be quite natural. It indicates how "true" a given proposition is. The value $\mathfrak{f}$ indicates that the proposition is definitely not true, $\mathfrak{u}$ admits a possibility that the proposition is true, $\mathfrak{i}$ shows that there

[1] For simplicity we use the same symbols to denote truth constants and corresponding truth values.

is at least one witness/evidence indicating the truth of the proposition, and finally, $\mathfrak{t}$ expresses that the proposition is definitely true.

Note that (1) linearizes the truth ordering of [4], where $\mathfrak{u}$ and $\mathfrak{i}$ are incomparable. This linearization is compatible with knowledge ordering of [4] where $\mathfrak{u} < \mathfrak{i}$.

According to [35], the pragmatics of disjunction should include the following principles:

- disjunction is true only when at least one of its operands is true;
- disjunction is false only when all its operands are false;

and the pragmatics of conjunction:

- conjunction is true only when all its operands are true;
- conjunction is false only when at least one of its operands is false.

The implication $\rightarrow$ is a four-valued extension of the classical implication. It is motivated and discussed in [22,23,37,35]. Observe that implication can only be $\mathfrak{t}$ or $\mathfrak{f}$. Implication

$$\textit{premisses} \rightarrow \textit{conclusion}$$

reflects the following principles [35]:

- truth or falsity of the conclusion can only be deduced when premisses are true;
- when premisses are inconsistent, conclusion should be inconsistent, too;
- false or unknown premisses do not participate in deriving new conclusions.

Remark 3.2. It is worth emphasizing that [35]:

- when one restricts truth values to $\{\mathfrak{t}, \mathfrak{f}\}$ then connectives defined in Table 1 become equivalent to their counterparts in classical propositional logic;
- when one restricts truth values to $\{\mathfrak{t}, \mathfrak{u}, \mathfrak{f}\}$ or to $\{\mathfrak{t}, \mathfrak{i}, \mathfrak{f}\}$ then conjunction, disjunction and negation become respectively their counterparts in Kleene three-valued logic K_3 with the third (non-classical) value meaning *undetermined* and in Priest logic P_3 [30], where the third value receives the meaning *paradoxical*.[2] ◁

Let $v : \textit{Var} \longrightarrow \textit{Const}$ be a *valuation of variables*. For a literal ℓ, by $\ell(v)$ we understand the ground literal obtained from ℓ by substituting each variable x occurring in ℓ by constant $v(x)$.

Definition 3.3. The *truth value* of a literal ℓ w.r.t. a set of ground literals L and valuation v, denoted by $\ell(L, v)$, is defined as follows:

$$\ell(L,v) \stackrel{\text{def}}{=} \begin{cases} \mathfrak{t} \text{ if } \ell(v) \in L \text{ and } (\neg\ell(v)) \notin L; \\ \mathfrak{i} \text{ if } \ell(v) \in L \text{ and } (\neg\ell(v)) \in L; \\ \mathfrak{u} \text{ if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \notin L; \\ \mathfrak{f} \text{ if } \ell(v) \notin L \text{ and } (\neg\ell(v)) \in L. \end{cases}$$ ◁

[2] The only difference between K_3 and P_3 is that only *true* is designated in K_3, while in P_3 both *true* and *paradoxical* are.

Table 2. Semantics of first-order formulas

- if α is a literal then $\alpha(L, v)$ is defined in Definition 3.3;
- $(\neg\alpha)(L, v) \stackrel{\text{def}}{=} \neg(\alpha(L, v))$, where $\neg$ at the righthand side of equality is defined in Table 1;
- $(\alpha \circ \beta)(L, v) \stackrel{\text{def}}{=} \alpha(L, v) \circ \beta(L, v)$, where $\circ \in \{\vee, \wedge, \rightarrow\}$;
- $(\forall x \alpha(x))(L, v) \stackrel{\text{def}}{=} \min_{a \in Const} \{(\alpha_a^x)(L, v)\}$, where min is the minimum w.r.t. ordering (1);
- $(\exists x \alpha(x))(L, v) \stackrel{\text{def}}{=} \max_{a \in Const} \{(\alpha_a^x)(L, v)\}$, where max is the maximum w.r.t. ordering (1).

Example 3.4. Consider the situation described in Example 2.1 and let $v(x) = low$. Then, for example, $P(a, x)(C_p, v) = \mathsf{t}$ and $P(a, x)(C_e, v) = \mathsf{u}$. ◁

For a formula $\alpha(x)$ with a free variable x and $c \in Const$, by $\alpha(x)_c^x$ we understand the formula obtained from α by substituting all free occurrences of x by c. Definition 3.3 is extended to all formulas in Table 2, where α and β denote first-order formulas, v is a valuation of variables, L is a set of ground literals, and the semantics of propositional connectives appearing at righthand sides of equivalences is given in Table 1.

Let us now define belief structures based on sets of literals. In this context the concept of an epistemic profile is the key abstraction involved in belief formation.

If S is a set then by $\text{FIN}(S)$ we understand the set of all finite subsets of S.

Let us now define the concepts of belief structures and epistemic profiles which are central to our approach.

Definition 3.5. Let $\mathbb{C} \stackrel{\text{def}}{=} \text{FIN}(\mathcal{G}(Const))$ be the set of all finite sets of ground literals over the set of constants *Const*. Then:

- by a *constituent* we understand any set $C \in \mathbb{C}$;
- by an *epistemic profile* we understand any function $\mathcal{E} : \text{FIN}(\mathbb{C}) \longrightarrow \mathbb{C}$;
- by a *belief structure over an epistemic profile* $\mathcal{E}$ we mean $\mathcal{B}^{\mathcal{E}} = \langle \mathcal{C}, F \rangle$, where:
  - $\mathcal{C} \subseteq \mathbb{C}$ is a nonempty set of constituents;
  - $F \stackrel{\text{def}}{=} \mathcal{E}(\mathcal{C})$ is the *consequent* of $\mathcal{B}^{\mathcal{E}}$. ◁

Example 3.6. For the Example 2.1,

$\mathcal{C} = \{C_p, C_n, C_e\}$ and $F = \{\neg S(a, healthy), S(a, healthy)\}$,

so $\mathcal{E}$ is any function of the signature required in Definition 3.5 such that $\mathcal{E}(\mathcal{C}) = F$. ◁

Note that constituents and consequents contain ground literals only. Of course, they can be defined using advanced theories or deductive database technologies. Therefore, if one wants to express beliefs as expressions more complex than just literals, it can be done. In Definition 3.5 we do not restrict representation of constituents or consequents. It may be of arbitrary complexity. The only requirement is that representations used should finally return finite sets of ground literals. The same applies to epistemic profiles. However, when tractability is to be achieved, such representations should be restricted to algorithms running in deterministic polynomial time. Our choice is to find representations based on deductive database technologies that ensure tractability and capture all polynomially computable queries. This way we guarantee both tractability and possibility of expressing all epistemic profiles and belief structures constructible in deterministic polynomial time.

Definition 3.7. Let $\mathcal{E}$ be an epistemic profile. The *truth value of formula* α w.r.t. belief structure $\mathcal{B}^{\mathcal{E}} = \langle \mathcal{C}, F \rangle$ and valuation v, denoted by $\alpha(\mathcal{B}^{\mathcal{E}}, v)$, is defined by:[3]

$$\alpha(\mathcal{B}^{\mathcal{E}}, v) \stackrel{\text{def}}{=} \alpha(\bigcup_{C \in \mathcal{C}} C, v). \qquad \lhd$$

Example 3.8. Consider again the situation described in Example 2.1. Let $v(x) = low$ and the belief structure $\mathcal{B}^{\mathcal{E}}$ be as described in Example 3.6. Then,

$$\bigcup_{C \in \mathcal{C}} C = \{P(a, \mathit{low}), N_1(a, \mathit{high}), \neg Cl(a, \mathit{noisy}), Cl(a, \mathit{neutral}), Cl(a, \mathit{pollutive})\}.$$

Therefore, e.g., $(P(a,x) \wedge N_1(a,x))(\mathcal{B}^{\mathcal{E}}, v) = \mathfrak{u}$ and $(P(a,x) \vee N_1(a,x))(\mathcal{B}^{\mathcal{E}}, v) = \mathfrak{t}$. Observe that truth of $N_1(a, high)$ does not automatically imply falsity of $N_1(a, low)$.$\lhd$

To express beliefs, we extend the language with operator $\mathrm{Bel}()$ standing for beliefs. The truth table for $\mathrm{Bel}()$ is:

$$\mathrm{Bel}(\mathfrak{t}) \stackrel{\text{def}}{=} \mathfrak{t}, \quad \mathrm{Bel}(\mathfrak{i}) \stackrel{\text{def}}{=} \mathfrak{i}, \quad \mathrm{Bel}(\mathfrak{u}) \stackrel{\text{def}}{=} \mathfrak{f}, \quad \mathrm{Bel}(\mathfrak{f}) \stackrel{\text{def}}{=} \mathfrak{f}. \tag{2}$$

We say that a formula is $\mathrm{Bel}()$-*free* if it contains no occurrences of the $\mathrm{Bel}()$ operator.

Definition 3.9. Let $\mathcal{E}$ be an epistemic profile. The *truth value of formula* α *w.r.t. belief structure* $\mathcal{B}^{\mathcal{E}} = \langle \mathcal{C}, F \rangle$ *and valuation* v, denoted by $\alpha(\mathcal{B}^{\mathcal{E}}, v)$, is defined as follows:

- clauses for propositional connectives and quantifiers are as in Table 2;
- when α is $\mathrm{Bel}()$-free then:
  - $\alpha(\mathcal{B}^{\mathcal{E}}, v)$ is defined by Definition 3.7;
  - $\mathrm{Bel}(\alpha)(\mathcal{B}^{\mathcal{E}}, v) \stackrel{\text{def}}{=} \mathrm{Bel}(\alpha(F, v))$, where the truth value $\alpha(F, v)$ is defined in Table 2 and $\mathrm{Bel}()$ applied to a truth value is defined by (2);
- when $\mathrm{Bel}()$ operators are nested in α then $\alpha(\mathcal{B}^{\mathcal{E}}, v)$ is evaluated starting from the innermost occurrence of $\mathrm{Bel}()$, which is then replaced by the obtained truth value, etc. $\lhd$

Example 3.10. For the belief structure $\mathcal{B}^{\mathcal{E}}$ introduced in Example 2.1 (see also Example 3.6) and $v(x) = \mathit{low}$, we have:

$$\begin{aligned}&\big(P(a,x) \wedge \mathrm{Bel}(P(a,x) \vee \mathrm{Bel}(S(a, \mathit{healthy})))\big)(\mathcal{B}^{\mathcal{E}}, v) = \\ &\quad \mathfrak{t} \wedge \mathrm{Bel}(P(a, \mathit{low}) \vee \mathrm{Bel}(\mathfrak{i})) = \mathfrak{t} \wedge \mathrm{Bel}(P(a, \mathit{low}) \vee \mathfrak{i}) = \mathfrak{t} \wedge \mathrm{Bel}(\mathfrak{u} \vee \mathfrak{i}) = \\ &\quad \mathfrak{t} \wedge \mathrm{Bel}(\mathfrak{i}) = \mathfrak{t} \wedge \mathfrak{i} = \mathfrak{i}. \qquad \lhd\end{aligned}$$

[3] Since $\bigcup_{C \in \mathcal{C}} C$ is a set of ground literals, $\alpha(\mathcal{S}, v)$ is well-defined by Table 2.

One can easily verify the following proposition.

Proposition 3.11. *For any formula α, belief structure $\mathcal{B}^{\mathcal{E}}$ and valuation of variables v:*

$$\big(\neg \mathrm{Bel}(\mathsf{f})\big)\big(\mathcal{B}^{\mathcal{E}}, v\big) = \mathsf{t}; \tag{3}$$

$$\big(\mathrm{Bel}(\alpha) \to \mathrm{Bel}(\mathrm{Bel}(\alpha))\big)\big(\mathcal{B}^{\mathcal{E}}, v\big) = \mathsf{t}; \tag{4}$$

$$\big(\neg \mathrm{Bel}(\alpha) \to \mathrm{Bel}(\neg \mathrm{Bel}(\alpha))\big)\big(\mathcal{B}^{\mathcal{E}}, v\big) = \mathsf{t}. \tag{5}$$

◁

Observe that the above formulas express the classical properties of beliefs: (3) is the axiom **D**, (4) and (5) are axioms **4** and **5**, expressing positive and negative introspection. Note that modal logic **KD45**, based on these axioms, is typically used to model beliefs in multiagent systems. Furthermore, there are belief structures, where the following axiom **T**, distinguishing knowledge and beliefs, does not have to be true:

$$\mathrm{Bel}(\alpha) \to \alpha. \tag{6}$$

This follows from the fact that given a belief structure $\langle \mathcal{C}, F \rangle$, $\mathrm{Bel}(\alpha)$ evaluates α in F while α itself is evaluated in $\bigcup \mathcal{C}$.

Let us also note that the following axiom:

$$\neg\big(\mathrm{Bel}(\alpha) \wedge \mathrm{Bel}(\neg\alpha)\big), \tag{7}$$

sometimes replacing axiom (3) is not always true under our semantics. For example, when α is $\mathfrak{i}$ then formula (7) is $\mathfrak{i}$. Axioms (3) and (7) are equivalent in the context of **KD45**. However, this is no longer the case in our semantics as we allow agents to have inconsistent beliefs.

Observe, however, that in our semantics $\mathrm{Bel}(\alpha \vee \beta)$ has always the same truth value as $\mathrm{Bel}(\alpha) \vee \mathrm{Bel}(\beta)$. This is caused by the fact that we assume that any epistemic profile delivers a single consequent (a single “world”). This is closer to the intuitionistic understanding of disjunction than to the classical one. In applications requiring that $\mathrm{Bel}(\alpha \vee \beta)$ does not force $\mathrm{Bel}(\alpha) \vee \mathrm{Bel}(\beta)$ one has to allow more than one consequent in Definition 3.5 introducing belief structures.

4 Pragmatics

4.1 Individual Beliefs

Agents can acquire knowledge about other agents' beliefs via communication and observation. In contrast to many existing approaches, we do not assume that an agent entering a group changes its beliefs. However, group beliefs prevail over individual ones. For example, when two agents cooperate, they may have certain beliefs as a group, but do not have to share them as individuals. Such a perspective usually results in a substantial improvement of complexity.

When the group is dismissed, agents continue to act according to their individual beliefs. These can be revised to reflect information acquired during cooperation. In our approach such revisions are delayed until the group disintegrates. Actually, in everyday life we frequently face similar situations. A group member does not need to share beliefs of group leaders, but still has to obey their commands. As group beliefs are typically built upon individual ones, immediate revisions of group members' beliefs could force revision at the group level. In reasonable cases one can expect that this process would converge to a fixpoint, but this is not guaranteed. Therefore, in certain situations infinite loops in belief revisions could occur.

When agents cooperate, a specific group is, possibly implicitly, created, including an epistemic profile fitting the entire situation. This is where belief fusion methods adequate for the group in question occur. In general, any interaction between agents leads to the creation of a (possibly virtual) group with a specific epistemic profile. We can naturally model this process in the proposed framework.

Let $\{Ag_i \mid i = 1, \ldots, n\}$ be a *set of agents*. To model individual beliefs we introduce belief operators $\mathrm{Bel}_i(\alpha)$, for $i = 1, \ldots, n$. As usually, the formula $\mathrm{Bel}_i(\alpha)$ expresses that agent Ag_i believes in α. To define the semantics of $\mathrm{Bel}_i(\alpha)$ operators we assume that for $i = 1, \ldots, n$, $\mathcal{E}_i$ is an epistemic profile of agent Ag_i and $\mathcal{B}^{\mathcal{E}_i} = \langle \mathcal{C}_i, F_i \rangle$ is a belief structure of agent Ag_i.

Definition 4.1. Let $\bar{\mathcal{B}} = \{\mathcal{B}^{\mathcal{E}_i} \mid i = 1, \ldots, n\}$ be a tuple of belief structures. The *truth value of formula α w.r.t. $\bar{\mathcal{B}}$ and valuation v w.r.t. agent Ag_i*, denoted by $\alpha(i, \bar{\mathcal{B}}, v)$, is defined as follows:

- clauses for propositional connectives and quantifiers are as in Table 2;
- when α is $\mathrm{Bel}()$-free then:
  - $\alpha(i, \bar{\mathcal{B}}, v)$ is defined as $\alpha(\mathcal{B}^{\mathcal{E}_i}, v)$ in the sense of Definition 3.7;
  - $\mathrm{Bel}_j(\alpha)(i, \bar{\mathcal{B}}, v) \stackrel{\mathrm{def}}{=} \mathrm{Bel}(\alpha(F_j, v))$, where the truth value $\alpha(F_i, v)$ is defined in Table 2 and $\mathrm{Bel}()$ applied to a truth value is defined by (2);
- when $\mathrm{Bel}_j()$ operators are nested in α then $\alpha(i, \bar{\mathcal{B}}, v)$ is evaluated starting from the innermost occurrence of $\mathrm{Bel}()$, which is then replaced by the obtained truth value, etc. $\triangleleft$

Example 4.2. When agent Ag_k evaluates formula $\big(r \vee \mathrm{Bel}_i(\mathrm{Bel}_j(p) \wedge q)\big)$ w.r.t. v then:

- r is evaluated w.r.t. F_k and v;
- p in $\mathrm{Bel}_j(p)$ is evaluated w.r.t. F_j and v;
- q in $\mathrm{Bel}_i(\mathrm{Bel}_j(p) \wedge q)$ is evaluated w.r.t. F_i and v. $\triangleleft$

4.2 Group Beliefs

A group of agents, say $G = \{Ag_{i_1}, \ldots, Ag_{i_k}\}$, has its *group belief structure* $\mathcal{B}^{\mathcal{E}_G} = \langle \mathcal{C}_G, F_G \rangle$, where $\mathcal{C}_G = \{F_{i_1}, \ldots, F_{i_k}\}$. Thus, consequents of group members become constituents of a group. The group then builds group beliefs via its epistemic profile $\mathcal{E}_G$, e.g. by adjudicating beliefs of group members, and reaches its consequent F_G (see Figure 1).

To express properties of group beliefs we extend the language by allowing operators $\mathrm{Bel}_G(\alpha)$, where G is a group of agents.

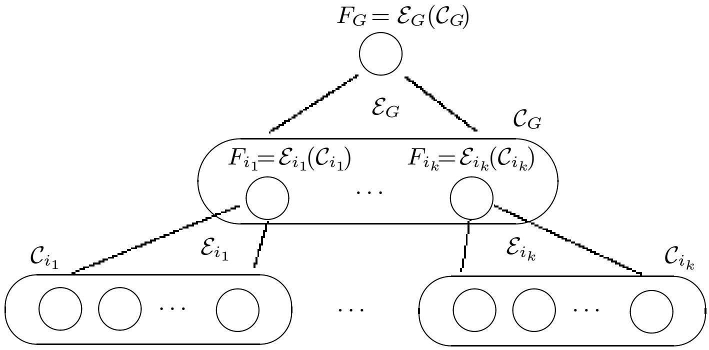

Fig. 1. The architecture of individual and group beliefs

Let $\{Ag_i \mid i = 1, \ldots, n\}$ be a *set of agents* and $\{G_j \mid j = n+1, \ldots m\}$ be a set of groups of agents. To define the semantics of $\mathrm{Bel}_G(\alpha)$ operators we extend Definition 4.1 by assuming that for $l = n+1, \ldots, m$, $\mathcal{E}_l$ is an epistemic profile of group G_l and $\mathcal{B}^{\mathcal{E}_l} = \langle \mathcal{C}_l, F_l \rangle$ is a belief structure of group G_l. We therefore have a tuple of belief structures $\bar{\mathcal{B}} = \{\mathcal{B}^{\mathcal{E}_l} \mid l = 1, \ldots, m\}$, where for $i = 1, \ldots, n$, $\mathcal{B}^{\mathcal{E}_i}$ is a belief structure of agent Ag_i and for $j = n+1, \ldots, m$, $\mathcal{B}^{\mathcal{E}_j}$ is a belief structure of group G_j.

Since groups are dealt with exactly as agents, given a tuple of belief structures $\bar{\mathcal{B}} = \{\mathcal{B}^{\mathcal{E}_l} \mid l = 1, \ldots, m\}$, the truth value of formula α w.r.t. $\bar{\mathcal{B}}$ and valuation v w.r.t. agent Ag_i (respectively, group G_i), is defined exactly as in Definition 4.1, assuming that indices $1, \ldots, n$ refer to agents and indices $n+1, \ldots, m$ refer to groups.

4.3 Other Complex Beliefs

The same way, beliefs of groups involving other groups may be formed. For example, a surveillance group of robots G_s may join a rescue team of robots G_r making a larger group $G_{s,r}$. Then the consequents of G_s and G_r become constituents of $G_{s,r}$. Furthermore, such groups can become parts of other, more complex groups, and so on. The underlying methods for forming group beliefs on the top of group members' beliefs are typically highly application- and context-dependent.

To express beliefs of such groups we extend the language with $\mathrm{Bel}_{\mathbb{G}}()$ operators, where $\mathbb{G}$ may contain individual agents, groups of agents, groups of groups of agents, etc. Since such complex groups, when formed, are equipped with belief structures like in the case of groups consisting of agents only, the semantics of $\mathrm{Bel}_{\mathbb{G}}()$ operators is given by an immediate adaptation of the semantics of groups.

Note that, due to complexity reasons, it is reasonable to assume that only formed groups are equipped with belief structures and epistemic profiles. When a group does not exist, we assume that its belief structure $\mathcal{B}$ is "empty", i.e., $\mathcal{B} = \langle \mathcal{C}, F \rangle$ with $\mathcal{C} = \{\emptyset\}$ (that is, $\mathcal{C}$ consists of a single set being the empty set) and $F = \emptyset$. Note that all queries supplied to this structure return the value $\mathfrak{u}$.

4.4 Querying Belief Structures

Traditional deductive databases are mainly based on the classical logic [1]. Belief operators are rather rarely considered in such contexts (but see, e.g., [9,28,29]). In our approach belief operators relate formulas to consequents which are sets of ground literals. Therefore, rather than with possible worlds, we always deal with sets of ground literals present in considered structures. Namely, in order to find out what are actual beliefs of agents, groups of agents, etc., the mechanism based on querying belief structures is applicable. For example one can ask the following queries:

$\mathrm{Bel}(\exists X(S(X, \mathit{healthy})))$ – is it believed that there is a healthy place?
$\mathrm{Bel}(\forall X(S(X, \mathit{healthy})))$ – is it believed that all places are healthy?

Belief fusion requires gathering beliefs of different agents. For example, the following query:

$$\mathrm{Bel}_1(\exists X(S(X, \mathit{healthy}))) \wedge \mathrm{Bel}_2(\exists X(S(X, \mathit{healthy}))), \tag{8}$$

allows us to check whether agents $Ag1$ and $Ag2$ believe that there is a place where the situation is healthy. Formula (8) contains no free variables, so the query returns a truth value. Of course, beliefs of these agents do not have to refer to the same place. If one intends to verify whether there is a place believed to be healthy by both agents simultaneously, then the query should rather be formulated as:

$$\exists X(\mathrm{Bel}_1(S(X, \mathit{healthy})) \wedge \mathrm{Bel}_2(S(X, \mathit{healthy}))). \tag{9}$$

Using query:

$$\mathrm{Bel}_1(\forall X(S(X, \mathit{healthy}))) \vee \mathrm{Bel}_2(\forall X(S(X, \mathit{healthy}))) \tag{10}$$

one can ask whether at least one of agents believes that all places are healthy.

On the other hand, query:

$$\forall X(\mathrm{Bel}_1(S(X, \mathit{healthy})) \vee \mathrm{Bel}_2(S(X, \mathit{healthy}))) \tag{11}$$

expresses the fact that every place is believed to be healthy by at least one agent.

When formulas used as queries contain free variables, queries return tuples together with an information whether a given tuple makes the query $\mathfrak{t}$, $\mathfrak{f}$ or $\mathfrak{i}$ (tuples making the query $\mathfrak{u}$ are not returned) – see [35]. For example, the query:

$$\mathrm{Bel}_1(S(X, \mathit{healthy})) \wedge \mathrm{Bel}_2(S(X, \mathit{healthy})) \tag{12}$$

asks for values of X such that both agents believe that X is healthy. One can get an answer that a place a is healthy, a place b is not healthy, that it is inconsistent that a place c is healthy. Since there is no information about other places, it is unknown whether these places are healthy or not.

Such, possibly rather complex, queries are naturally used in designing epistemic profiles. Let us also note, that in multiagent settings they provide a powerful mechanism for deciding which actions to perform.

5 Distributed Beliefs

In contemporary intelligent distributed systems, like multiagent systems, we typically deal with many heterogenous information sources. They independently deliver information (e.g. percepts), expressed in terms of beliefs, on various aspects of a recent situation. Depending on the context and the goal of the reasoning process, different beliefs need to be fused in order to achieve more holistic judgement of the situation. Apparently, this information fusion may be realized in various ways. Let us now take a closer look at this formal process.

Distributed information sources naturally introduce four truth values [4,10,22,23,24]. On the other hand, in real-world distributed problem solving, lack of knowledge and inconsistent beliefs are to be resolved at some point. More precisely, in the case of lacking knowledge, we need to complete missing information at the objective level, while in the case of inconsistencies we do this at a meta-level, for example, by verifying which information sources deliver false information. This knowledge can then be used for a better setup or calibration of sensors and other data sources, as well as diagnostic systems detecting malfunctioning devices. When this is impossible, especially in time-critical systems, commonsense reasoning methods can be of help as they generally characterize typical situations [6,21,25]. Among these methods (local) closed world assumption, default reasoning, autoepistemic reasoning and defeasible reasoning are of primary importance. Again, 4QL supports such forms of reasoning.

As we shall discuss in the next sections, using our approach one can achieve a tractable model of distributed belief fusion, as well as an implementation framework of distributed belief fusion via epistemic profiles.

6 Towards Implementation

As indicated before, we apply reasoning over databases rather than over general theories. Such an approach reflects the reality of intelligent systems and significantly reduces the complexity of reasoning, typically from at least exponential to deterministic polynomial time, no matter what type of reasoning is used. This substantial complexity gain is achieved by using 4QL, a query language [22,23,24] which enjoys tractable query computation and captures all tractable queries. Belief structures, when implemented in 4QL, can be considered as sets of ground literals generated by facts and rules. Therefore, one can tractably query belief structures using such query languages as first-order queries, fixpoint queries or 4QL queries.

6.1 The Main Idea

The main idea is illustrated in Figure 2. Namely, we propose to implement epistemic profiles via an intermediate layer consisting of *derivatives*, where each derivative is a finite set of ground literals. Intuitively, derivatives represent intermediate belief fusion results or, in other words, intermediate views on the situation in question. Importantly, such a structure allows us to implement belief fusion in a highly distributed manner.

Example 6.1. One can consider two derivatives:

- D_p – for deciding the pollution level;
- D_n – for deciding the noise level.

Such derivatives should result from reasoning patterns defined by the corresponding epistemic profile of the considered agent. For example, these derivatives may be:

$$D_p = \{P(a, \mathit{moderate})\}, \;\; D_n = \{N(a, \mathit{high})\}.$$

Based on the contents of D_p and D_n, the agent has to decide whether the situation is healthy or not (and include it in its set of consequents F). ◁

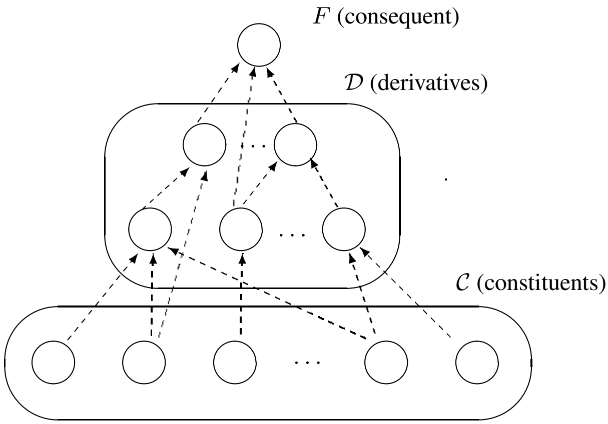

Fig. 2. Implementation framework for belief structures and epistemic profiles. Arrows indicate belief fusion processes.

6.2 Implementation Tool: 4QL

There are several languages designed for programming BDI agents (for a survey see, e.g., [27]). However, none of these approaches directly addresses belief formation, in particular nonmonotonic/defeasible reasoning techniques. Our choice is therefore 4QL, a DATALOG$^{\neg\neg}$-like query language. It supports a modular and layered architecture, and provides a tractable framework for many forms of rule-based reasoning both monotonic and nonmonotonic. As the underpinning principle, openness of the world is assumed, which may lead to the lack of knowledge. Negation in rule heads, expressing negative conclusions, may lead to inconsistencies. As indicated in [22], to reduce the unknown/inconsistent zones, *modules* and *external literals* provide means for:

- the application-specific disambiguation of inconsistent information;
- the use of (Local) Closed World Assumption;
- the implementation of various forms of nonmonotonic and defeasible reasoning.

To express nonmonotonic/defeasible rules we apply modules as well as external literals, originally introduced in [22]. Importantly, different modules can be distributed among different agents participating in the reasoning process.

In the sequel *Mod* denotes the set of *module names*.

Definition 6.2. An *external literal* is an expression of one of the forms:

$$M.R, -M.R, M.R \text{ IN } T, -M.R \text{ IN } T, \tag{13}$$

where $M \in Mod$ is a module name, R is a positive literal, '$-$' stands for negation and $T \subseteq \{\mathfrak{f}, \mathfrak{u}, \mathfrak{i}, \mathfrak{t}\}$. For literals (13), module M is called the *reference module*. ◁

The intended meaning of "$M.R$ IN T" is that the truth value of $M.R$ is in the set T. External literals allow one to access values of literals in other modules. If R is not defined in the module M then the value of $M.R$ is assumed to be $\mathfrak{u}$.

Assume a strict tree-like order $\prec$ on *Mod* dividing modules into layers. An external literal with reference module M_1 may appear in rule bodies of a module M_2, provided that $M_1 \prec M_2$.[4]

Definition 6.3. By a *rule* we mean any expression of the form:

$$\ell :- b_{11}, \ldots, b_{1i_1} \mid \ldots \mid b_{m1}, \ldots, b_{mi_m}. \tag{14}$$

where ℓ is a literal, $b_{11}, \ldots, b_{1i_1}, \ldots, b_{m1}, \ldots, b_{mi_m}$ are literals or external literals, and ',' and '|' abbreviate conjunction and disjunction, respectively.

Literal ℓ is called the *head* of the rule and the expression at the righthand side of :– in (14) is called the *body* of the rule. ◁

Rules of the form (14) are understood as implications:

$$((b_{11} \wedge \ldots \wedge b_{1i_1}) \vee \ldots \vee (b_{m1} \wedge \ldots \wedge b_{mi_m})) \rightarrow \ell,$$

where it is assumed that the empty body takes the value $\mathfrak{t}$ in any set of literals.

By convention, facts are rules with the empty body. For example, a fact '$P(a)$.' is an abbreviation for the rule '$P(a)$:– .' .

Definition 6.4. Let a set of constants, *Const*, be given. A set of ground literals L with constants in *Const* is a *model of a set of rules* S iff each ground instance of each rule of S (understood as implication) obtains the value $\mathfrak{t}$ in L. ◁

The semantics of 4QL is defined via well-supported models generalizing the idea presented in [14]. Intuitively, a model is *well-supported* if all derived literals are supported by a reasoning grounded in facts. It appears that for any set of rules there is a unique well-supported model and it can be computed in polynomial time. For details see [24].

Remark 6.5. One can further extend 4QL without losing its tractability by allowing arbitrary first-order formulas in bodies of rules. This allows one to directly implement queries like those considered in Section 4.4. For details of such an extension see [35].◁

[4] Observe that layers generalize the concept of stratification of DATALOG$^\neg$ queries [23] (for definition of stratification see, e.g., [1]).

6.3 Implementing Belief Structures in 4QL

Implementation of tractable belief structures and epistemic profiles is now relatively easy. Namely,

- epistemic profiles can be implemented by the use of derivatives;
- every constituent, derivative and consequent can be implemented as a separate 4QL module.

The hierarchical structure of derivatives makes it possible to use the layered architecture of modules, as required in 4QL. External literals allow to access information to create beliefs on the basis of perhaps still preliminary beliefs already obtained.

Let $\{\mathcal{E}_i \mid i = 1, \ldots, n\}$ be epistemic profiles of agents $Ag_1, \ldots, Ag_n$ and let $\mathcal{B}_i^{\mathcal{E}} = \langle \mathcal{C}_i, F_i \rangle$ be the agents' belief structures over these epistemic profiles. Assume an agent Ag_k ($1 \leq k \leq n$) is asked a query. We have the following two cases:

1. when a formula expressing (a part of) a query is not within the scope of a $\mathrm{Bel}()$ operator then we evaluate it in the database obtained as the union of constituents $\bigcup_{C \in \mathcal{C}_k} C$ (according to Definition 3.7);
2. when a formula has the form $\mathrm{Bel}_j(\alpha)$ ($1 \leq j \leq n$), it is evaluated in F_j (considered as a database).

Example 6.6. Consider queries (8)– (11) (Section 4.4). If $Ag_1.S, Ag_2.S$ respectively refer to relation S included in the set of consequents of Ag_1's and Ag_2's belief structures then queries (8)– (11) can be expressed by:

$$\begin{array}{l}
\exists X(Ag_1.S(X, \textit{healthy})) \wedge \exists X(Ag_2.S(X, \textit{healthy})),\\
\exists X(Ag_1.S(X, \textit{healthy}) \wedge Ag_2.S(X, \textit{healthy})),\\
\forall X(Ag_1.S(X, \textit{healthy})) \vee \forall X(Ag_2.S(X, \textit{healthy})),\\
\forall X(Ag_1.S(X, \textit{healthy}) \vee Ag_2.S(X, \textit{healthy})),
\end{array}$$

where $Ag_i.S$ indicates that the value of S is taken from consequents of agent Ag_i. $\triangleleft$

Open source interpreters [32,33] of 4QL are available via `4ql.org`. Also, a commercial implementation of 4QL is being recently developed by NASK.[5]

6.4 Exemplary Implementation

Consider now the scenario outlined in Examples 2.1 and 6.1. Exemplary modules corresponding to constituents, derivatives and consequents are shown in Tables 3–5, respectively.[6]

It is important to note that well-supported models are sets of literals. Thus relational or deductive databases technology can be used to query them (see also Section 4.4).

In fact, four logical values, external literals and modular architecture distinguish 4QL from many other approaches (for a survey see, e.g., [3]). 4QL modules are structured

[5] `http://www.nask.pl/nask_en/`

[6] We use the Inter4QL self-explanatory syntax – for details see [33].

Table 3. Modules corresponding to constituents considered in Example 2.1

`module Cp:`
`  domains:`
`    literal level.`
`    literal place.`
`  relations:`
`    P(place, level).`
`  facts:`
`    P(a, low).`
`end.`
	`module Cn:`
	`  domains:`
	`    literal level.`
	`    literal place.`
	`  relations:`
	`    N1(place, level).`
	`    N2(place, level).`
	`  facts:`
	`    N1(a, high).`
		`module Ce:`
		`  domains:`
		`    literal level.`
		`    literal place.`
		`  relations:`
		`    Cl(place, type).`
		`  facts:`
		`    -Cl(a, noisy).`
		`    Cl(a,neutral).`
		`    Cl(a,pollutive).`
		`end.`

into layers. According to syntax of 4QL, the lowest layer represents monotonic reasoning, while higher ones allow the user to provide (nonmonotonic) rules for disambiguating inconsistencies and completing lacking information. This is achieved by explicitly referring to logical values via external literals.

6.5 Complexity

Theoretically, any belief structure (also augmented with derivatives) can be of exponential size w.r.t. the number of literals involved. In applications they may be dynamically generated as agents appear or groups are formed. However, at a given timepoint we can assume that their size is always reasonable, as it reflects available resources. Also the number of groups may, in theory, be of exponential size w.r.t. the number of agents. Again, in a given application we have a limited number of groups and only these are equipped with belief structures, as discussed in [10] and in this paper itself.

Let $|Const| = k$, let n be the number of agents and let m be the number of groups. Further, let the size of all belief structures involved be bounded by $f(k, n, m)$. Then we have the following theorem which follows from the tractability of 4QL [22,23,24].

Theorem 6.7. *If belief structures and queries are implemented using the 4QL query language then the time complexity of computing queries is deterministic polynomial in* $f(k, n, m)$. ◁

Let us emphasize again that in practice one can safely assume that $f(k, n, m)$ is bounded by available resources, including time, memory and physical devices.

Theorem 6.7 holds also when 4QL is replaced by any query language with polynomially bounded complexity of computing queries. However, 4QL captures all polynomially computable queries [23], so when 4QL is used, we also have another theorem.

Theorem 6.8. *Any polynomially constructed belief structure can be implemented using 4QL.* ◁

It is worth emphasizing that Theorem 6.8 shows that 4QL is a sufficient language that serves our purposes. One can argue that the same applies to any query language which captures deterministic polynomial time. However, in contrast to other languages,

Table 4. Modules corresponding to derivatives considered in Example 6.1

```
module Dp:
  domains:
    literal level.
    literal place.
  relations:
    P(place, level).
  rules:
    P(X, moderate):-  Cp.P(X, low) IN {TRUE, UNKNOWN},
                      Ce.Cl(X, pollutive) IN {TRUE, UNKNOWN}.
      ...
end.

module Dn:
  domains:
    literal level.
    literal place.
  relations:
    N(place, level).
  rules:
    N(X,Y):- Cn.N1(X,Y), Cn.N2(X,Y) IN {TRUE, UNKNOWN} |
             Cn.N1(X,Y) IN {TRUE, UNKNOWN}, Cn.N2(X,Y).
      ...
end.
```

Table 5. Module corresponding to consequents considered in Example 2.1

```
module F:
  domains:
    literal characteristics.
    literal place.
  relations:
    S(place, characteristics).
  rules:
    -S(X, healthy):- Dn.N1(X, high), Dn.N2(X, high).
    S(X, healthy):- Cp.P(X, moderate),
                    Cn.N1(X, low) IN {TRUE, UNKNOWN}.
       ...
end.
```

4QL provides simple, but powerful tools for direct expression of a wide spectrum of reasoning techniques, including nonmonotonic ones, and allowing one to handle inconsistencies.

Observe also that agents' and groups' belief structures and epistemic profiles typically match some patterns reflecting agents' types or groups' organizational structures and cooperation procedures. Therefore, in practice, one can expect belief structures to be generated on the basis of a library of patterns, much like in object-oriented programming dynamic objects are generated on the basis of static classes, developed during the system's design phase. Of course, the number of such patterns does not change during system execution, so can be bound by a constant.

7 Conclusions

In this paper we differentiated agents' characteristics via individual and group epistemic profiles, reflecting agents' reasoning capabilities. This abstraction tool permits both to flexibly define the way an agent (a group) reasons and to reflect the granularity of reasoning. The presented pragmatic framework to beliefs suits real-world applications that often are not easy to formalize. In particular, it allows for natural handling of inconsistencies and gaps in beliefs by using paraconsistent and nonmonotonic reasoning.

Moreover, our approach permits a uniform modeling of individual and group beliefs, where group is a generic concept consisting of individual agents, groups of agents, groups of groups of agents, etc. Importantly, the assumed layered architecture underlying the framework allows one to avoid costly revisions of agents' beliefs when they join a group. This is especially important when paradigmatic agent interactions are considered. Cooperation, coordination and communication is naturally modeled by creating a group and forming group beliefs to achieve a common informational stance. What sort of structure it is and how this influences agents' individual beliefs is a matter of design decisions. Our approach ensures both the heterogeneity of agents involved and flexibility of group level reasoning patterns.

Most significantly, we have indicated 4QL as a tool to implement all epistemic profiles and belief structures constructible in deterministic polynomial time. We have also shown a natural methodology to obtain such implementations. One can then query implemented belief structures in a tractable manner, which provides a rich but still pragmatic reasoning machinery. To the best of our knowledge, such tractability of reasoning about beliefs has not been achieved yet. Also, nonmonotonic/defeasible reasoning techniques are easily expressible in 4QL, ensuring both richness and flexibility of implemented epistemic profiles.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley Pub. Co. (1996)
2. Ågotnes, T., Alechina, N.: Full and relative awareness: A decidable logic for reasoning about knowledge of unawareness. In: Proc. of TARK, pp. 6–14. ACM Press (2007)
3. Alferes, J.J., Pereira, L.M.: Reasoning with Logic Programming. LNCS, vol. 1111. Springer, Heidelberg (1996)
4. Belnap, N.D.: A useful four-valued logic. In: Epstein, G., Dunn, J.M. (eds.) Modern Uses of Many Valued Logic, pp. 8–37. Reidel (1977)
5. Béziau, J.-Y., Carnielli, W., Gabbay, D.M. (eds.): Handbook of Paraconsistency. College Publications (2007)
6. Brewka, G.: Non-Monotonic Reasoning: Logical Foundations of Commonsense. Cambridge University Press (1991)
7. de Amo, S., Pais, M.S.: A paraconsistent logic approach for querying inconsistent databases. International Journal of Approximate Reasoning 46, 366–386 (2007)
8. Dubois, D.: On ignorance and contradiction considered as truth-values. Logic Journal of the IGPL 16(2), 195–216 (2008)
9. Dunin-Kęplicz, B., Nguyen, L.A., Szałas, A.: Tractable approximate knowledge fusion using the Horn fragment of serial propositional dynamic logic. Int. J. Approx. Reasoning 51(3), 346–362 (2010)

10. Dunin-Kęplicz, B., Szałas, A.: Epistemic Profiles and Belief Structures. In: Jezic, G., Kusek, M., Nguyen, N.-T., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2012. LNCS, vol. 7327, pp. 360–369. Springer, Heidelberg (2012)
11. Dunin-Kęplicz, B., Szałas, A.: Paraconsistent Distributed Belief Fusion. In: Proc. IDC 2012, 6th International Symposium on Intelligent Distributed Computing. SCI, vol. 446, pp. 59–69. Springer, Heidelberg (2012)
12. Dunin-Kęplicz, B., Verbrugge, R.: Teamwork in Multi-Agent Systems. A Formal Approach. John Wiley & Sons, Ltd. (2010)
13. Dunin-Kęplicz, B., Verbrugge, R.: Awareness as a vital ingredient of teamwork. In: Stone, P., Weiss, G. (eds.) Proc. of AAMAS 2006, pp. 1017–1024 (2006)
14. Fages, F.: Consistency of Clark's completion and existence of stable models. Methods of Logic in Computer Science 1, 51–60 (1994)
15. Fagin, R., Halpern, J.: Belief, awareness, and limited reasoning. Artificial Intelligence 34(1), 39–76 (1988)
16. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning About Knowledge. The MIT Press (2003)
17. Gottlob, G.: Complexity results for nonmonotonic logics. Journal of Logic and Computation 2(3), 397–425 (1992)
18. Hintikka, J.: Knowledge and Belief. Cornell University Press (1962)
19. Kraus, S., Lehmann, D.: Knowledge, belief and time. Theoretical Computer Science 58, 155–174 (1988)
20. Laux, A., Wansing, H. (eds.): Knowledge and Belief in Philosophy and Artificial Intelligence. Akademie Verlag, Berlin (1995)
21. Łukaszewicz, W.: Non-Monotonic Reasoning - Formalization of Commonsense Reasoning. Ellis Horwood (1990)
22. Małuszyński, J., Szałas, A.: Living with inconsistency and taming nonmonotonicity. In: de Moor, O., Gottlob, G., Furche, T., Sellers, A. (eds.) Datalog 2010. LNCS, vol. 6702, pp. 384–398. Springer, Heidelberg (2011)
23. Małuszyński, J., Szałas, A.: Logical foundations and complexity of 4QL, a query language with unrestricted negation. Journal of Applied Non-Classical Logics 21(2), 211–232 (2011)
24. Małuszyński, J., Szałas, A.: Partiality and Inconsistency in Agents Belief Bases. In: Agents and Multi-agent Systems – Technologies and Applications. IOS Press (to appear, 2013)
25. Marek, V.W., Truszczyński, M.: Nonmonotonic Logic. Springer (1993)
26. Mares, E.D.: A paraconsistent theory of belief revision. Erkenntnis 56, 229–246 (2002)
27. Mascardi, V., Demergasso, D., Ancona, D.: Languages for programming BDI-style agents: an overview. In: Corradini, F., De Paoli, F., Merelli, E., Omicini, A. (eds.) WOA 2005 - Workshop From Objects to Agents, pp. 9–15 (2005)
28. Nguyen, L.A.: Foundations of modal deductive databases. Fundamenta Informaticae 79(1-2), 85–135 (2007)
29. Nguyen, L.A.: Constructing finite least Kripke models for positive logic programs in serial regular grammar logics. Logic Journal of the IGPL 16(2), 175–193 (2008)
30. Priest, G.: Paraconsistent belief revision. Theoria 67(3), 214–228 (2001)
31. Rantala, V.: Impossible worlds semantics and logical omniscience. Acta Philosophica Fennica 35, 106–115 (1982)
32. Sobczyk, Ł.: 4QL Runner developer description (2012), `http://4ql.org/wp-content/uploads/2012/10/4qlRunner.pdf`
33. Spanily, P.: The Inter4QL interpreter (2012), `http://4ql.org/wp-content/uploads/2012/10/inter4ql.pdf`

34. Stalnaker, R.: On logics of knowledge and belief. Philosophical Studies 128, 169–199 (2006)
35. Szałas, A.: How an agent might think. Logic Journal of the IGPL 21(3), 515–535 (2013)
36. van Ditmarsch, H.P., van der Hoek, W., Kooi, B.P.: Dynamic Epistemic Logic. Springer (2007)
37. Vitória, A., Małuszyński, J., Szałas, A.: Modeling and reasoning with paraconsistent rough sets. Fundamenta Informaticae 97(4), 405–438 (2009)
38. Wansing, H.: A general possible worlds framework for reasoning about knowledge and belief. Studia Logica 49, 523–539 (1990)