Deciding between Conflicting Influences

Andreas Schmidt Jensen

Department of Applied Mathematics and Computer Science,
Technical University of Denmark, Kongens Lyngby, Denmark
ascje@dtu.dk

Abstract. This paper investigates an approach of decision making internally in an agent where a decision is based on preference and expectation. The approach uses a logic for qualitative decision theory proposed by Boutilier to express such notions. To make readily use of this we describe a simple method for generating preference and expectation models that respect certain rules provided by the agents, and we briefly discuss how to integrate the approach into an existing agent programming language.

1 Introduction

Agents taking part in a multi-agent system are usually seen as intelligent entities that autonomously are able to bring about (from their own perspectives) desirable states. The designer is in a fixed setting with a controlled number of agents and globally desirable states often able to implement the agents such that their own desirable states coincide with the globally desirable states. In open societies, agents often come from different sources and their desires cannot as such be assumed to match the global desires. A suggestion is to impose an organization on the agents, which can influence the actions of the agent toward the desires of the organization.

When agents are constrained by an organization, their own goals may conflict with those of the organization and they need in such cases to be able to decide which of the conflicting goals to pursue. In some of the previous work toward resolving such conflicts, desires and obligations are ordered a priori, so that an agent either prefers desires over obligations or obligations over desires. This results in agents that are always selfish (considering own goals more important than organizational goals) or always social (vice versa). We argue in this paper that such distinction can be too hard; even a selfish agent could in some cases benefit from preferring certain obligations to its desires. We consider an approach on how to resolve such (and other) conflicts, based on work in the area of qualitative decision theory by Boutilier [4], where the expected consequences of bringing about a state are considered. We show that this result in agents that are not always either social or selfish, but instead are able to decide based on the consequences of bringing about a state.

Our focus is on a general approach toward deciding between different kinds of influences, with the aim to show that although agents are subject to influences

M. Cossentino et al. (Eds.): EMAS 2013, LNAI 8245, pp. 137–155, 2013.
© Springer-Verlag Berlin Heidelberg 2013

from different entities, they are able to make decisions based on the current situation, their preferences and the expected consequence of bringing about a state. We do not focus explicitly on the choice between an agent's desires and the obligations from an organization, but emphasize that the approach is useful in this situation and other situations as well.

To make the approach readily useful we furthermore describe a simple method for generating models for preference and expectation based on basic rules, such as "I prefer to drive to work when it rains", specified by the agents.

The paper is organized as follows: In section 2, we discuss the issues that arise when an agent has to make a decision between conflicting influences. In section 3, we present a new approach on how to solve such conflicts without having to put the agents into the categories "selfish" or "social". We present a method for generating models that conform to the agent's preferences in section 4. In section 5, we discuss a case in which agents have conflicting influences and show that our method enables them to make a decision using their own preference and the expected consequence of bringing about each state. We briefly discuss how to implement the system and integrate it in an existing agent programming language in section 6. Finally, we conclude our work and discuss future research directions in section 7.

2 Conflicting Influences

Agents entering an environment will be subject to influences from multiple sources: their own desires, requests from other agents, and obligations from an organization. In the well-known BDI model, an agent's desires become intentions, when the agent commits to bringing about these desires. One could argue that if an agent wants to accept requests from other agents, or if it wants to adhere to the obligations of an organization, these influences are merely desires as well, i.e. the agent simply desires to do so. The incentives for doing so are however not clear, since there should be different reasons for committing to actual desires and to requests or obligations "disguised" as desires. For example, if an agent has a desire to move a box from A to B, it typically *wants* to do so. However if the agent wants to pay a bill before its due date, this "desire" has more likely arisen from the fact that the agent does not want to pay a fine, rather than being an actual desire to pay the bill. In such a situation, the desire may actually be an obligation or a request to pay the bill, which means that the agent should reason differently since the actual desire is to avoid paying a fine.

Furthermore, consider an agent that receives an undesirable request from an agent that it desires to help. It may choose to commit to the task even though the task itself is not desirable, because the desire to help the other agent is stronger than the desire not to perform the task (the consequence of *not* helping the other agent might be a bad reputation). Similarly, if an agent is obligated to perform certain tasks for an organization, it should not only be able to consider whether the task is desirable, but also weigh this against the penalty for violating the obligation.

We call something that the agent might choose to intend to do a "decision influence" rather than a desire since it, as argued above, may stem from many different sources rather than being merely desires. The agent naturally has to consider its desires since it would be irrational to ignore them, but the consequence of not reasoning about e.g. obligations might be intolerable so these influence the agent as well. This also means that the agent is not supposed to be reasoning explicitly about whether it should commit to bringing about an arbitrary obligation or desires; they are merely considered influences, and the agent is not concerned with the different types of influences: only the fact that they affect the decision process matters. Several approaches are proposed on how to let agents choose between specific types of influences (typically obligations and desires) [2,5,6,7,8], so we briefly discuss how our approach differs.

In [5] conflicts between beliefs, obligations, intentions and desires are discussed with a focus on a distinction between *internal* conflicts, e.g. contradictory beliefs, conflicting obligations and *external* conflicts, such as a desire which is in conflict with an obligation. The solution proposed, the BOID architecture, imposes a strict ordering between beliefs, obligations, intentions and desires, such that the order of derivation determines the agent's attitude. Thus different agent types emerge; an agent deriving desires before beliefs is a wishful-thinking agent, while an agent deriving obligations before desires is social.

We believe this ordering is too strong; if an agent is social, it will always choose obligations over desires, and vice versa for selfish agents. This might not always be appropriate. For instance, a selfish agent might desire not to go to work, but if the consequence of not fulfilling the obligation of going to work is severe (i.e. getting fired), even a selfish agent should consider this consequence before deciding not to go to work.

Dignum *et al.* suggests that *"both norms and obligations should be explicitly used as influences on an agent's behavior"* [7]. They represent obligations (and norms) using Prohairetic Deontic Logic [10], a preference-based dyadic deontic logic which allows for contrary-to-duty obligations (obligations holding in a subideal context). Furthermore, they propose a modified BDI-interpreter in which selected events are augmented with potential deontic events, which, put simply, are obligations and norms that may become applicable when choosing a plan. For instance, if agent a has an obligation to perform a task for agent b, and a does not intend to do so he ought to inform b about this. The modified interpreter generates a number of options depending on these potential events and chooses a relevant plan based on the agent's attitude.

In [8] it is argued that the preference orderings induced by desires, obligations and norms should be combined into a single ordering. It is noted that a common way to do so is to allow that a single preference ordering determine the aggregate ordering, such that the agent might always put obligations over norms and norms over desires, similarly to the BOID architecture. Another approach is also discussed in which the orderings are mapped into a common scale, such that very desirable situations could outweigh the cost of violating certain obligations. Such ordering should be quite dynamic since, for example, obligations toward a trusted agent should become less important if that agent becomes less trustworthy. Simple rules are presented to deal with few alternatives, but it is noted that the situation is more complex if an agent has to choose between three or more alternatives and none of the three orderings agree on a preferred alternative. A simple rule which orders the alternatives in a fixed order results in a very simple-minded agent and it is suggested that the consequences of different situations is considered, however this is not investigated further.

Different types of role enactments are identified in [6] and they describe an approach for verification of consistency of agent goals and role goals. They work with agents and roles in which goals are prioritized using an ordering and investigate what is required to make agents and roles are compatible. This leads to role enacting agents that can prioritize own goals and role goals in a combined ordering, thus not necessarily making agents explicitly selfish or social. They define different enactment types, such as selfish enactment in which the agent includes both own goals and role goals, but gives priority to own goals, and social enactment in which priority is given to the rule goals.

2.1 Consequence-Based Decisions

Performing an action will in many cases result in one or more side effects that may or may not be desirable for the agent performing the action. These side effects are part of the consequences of performing the action, and the agent can reason using more information by considering these consequences, thus enabling it to make better decisions. This suggests that in order to reason about bringing about a certain state, the agent should consider what consequences are expected when bringing about that state.

We therefore suggest that the agent should reason about the expected *consequences* of choosing to commit to a decision influence and furthermore that this reasoning should be based on both *preference* and *tolerance*. We use preference for influences and tolerance for expected consequences of influences, and the reason for using tolerance instead of preference in the case of consequences is that the agent should not need to desire the consequences of bringing about a state. Since the consequences are merely side effects, they need not be desired in the same way as the influences are. If a consequence is preferable, then clearly it is also tolerable but the opposite need not be the case (the agent might tolerate going to work even though *prefers* to stay at home). We define a situation as being tolerated when the opposite is not preferred (e.g., working is tolerated if staying at home is not preferred over working). Using the influences, we can build two sets to base a decision on: the set of most preferred influences, *Pref*, and the set of influences with the most tolerable expected consequences, *Tol*. We can identify different strategies for how to make a decision based on these sets, such as considering one set before the other or by using a combination of the sets:

$$Pref > Tol \tag{1}$$

$$Pref < Tol \tag{2}$$

$$Pref \cup Tol \tag{3}$$

$$Pref \cap Tol \tag{4}$$

We can let preferences take precedence (1) such that if a single influence is most preferred (in *Pref*) it is chosen, and only in case of multiple most preferred influences will tolerance be taken into account, or we can let tolerance take precedence (2), which gives the opposite situation. However, this means that only in some cases are both preference and tolerance taken into account, so an agent could choose to commit to something it prefers which leads to an intolerable situations that could have been avoided if both sets were taken into account. We could take the collective influences (3), but then the agent would have to choose between things it prefer and things it tolerate even though the former may be intolerable and the latter could be unwanted. Instead we could let the decision be the influences that are *both* preferred and tolerated (4), thus ensuring that the decision is preferred by the agent and that the consequences can be tolerated. In certain situations, these sets may not coincide, and we argue then that the safest decision is to choose something tolerated, since then even though the influence might not be *most* preferred, at least it will not lead to an intolerable state. Our approach makes use of the last strategy, i.e. taking the intersection of the sets, since it incorporates both measures in all situations, while not resulting in intolerable preferred states.

Note that our approach does not incorporate an explicit notion of organizations; the focus is on many different kinds of influences including the obligations toward an organization. As a result, we model consequences as expectations from the environment, that is, which possible world is the most expected, which is the second most and so on. This means that if the consequences of the violation of an obligation (i.e. sanctions) are specified in an organizational model, these consequences are in our approach modeled such that worlds in which the violation has occurred *and* a sanction has been imposed are more expected than the worlds where the violation has occurred without the agent being sanctioned. This will be evident in the example in section 5 where all expected consequences are incorporated into the same model.

3 Modeling Influence and Consequence

We base our work on the Logic for Qualitative Decision Theory (QDT) by Boutilier [4]. We briefly describe the semantics of QDT and define a few new abbreviations to be used in the decision-making.

The basic idea behind the QDT model is as follows. An agent has the ultimate desire of achieving the goals to which it is committed. This can be modeled by a possible worlds-model in which the agent has achieved its goal when it is in a world where those goals hold. The most preferred world in an ideal setting is the world in which the agent has achieved all of its goals. However, such world is often unreachable since the agent could have contradicting goals, other agents could prevent the agent from achieving all of its goals, an organization could impose obligations, which contradict the agent's goals, etc. By ordering the worlds in a preference relation, it is possible to choose the most preferred world(s) in a sub-ideal situation.

In order to decide between influences the consequence of bringing about a state should be taken into account. If the consequence of pursuing a personal desire is to be fired from your workplace, it might not be reasonable to do so even though the desire was more preferred than the obligations from work. We briefly describe QDT below before moving on to modeling the expected consequence of bringing about a state.

A QDT model is of the form:

$$M = \langle W, \leq_P, \leq_N, \pi \rangle,$$

where W is the non-empty set of worlds, $\leq_P$ is the transitive, connected preference ordering[1], $\leq_N$ is the transitive, connected normality ordering, and π is the valuation function. The normality ordering is used to model how likely each world is, e.g. it is normally cold when it is snowing, and the preference ordering is used to model an agent's preferences.

The semantics are as follows:

$$\begin{aligned}
M, w \models p &\iff p \in \pi(w) \\
M, w \models \neg\varphi &\iff M, w \not\models \varphi \\
M, w \models \varphi \wedge \psi &\iff M, w \models \varphi \wedge M, w \models \psi \\
M, w \models \Box_P\, \varphi &\iff \forall v \in W, v \leq_P w, M, v \models \varphi \\
M, w \models \overleftarrow{\Box}_P\, \varphi &\iff \forall v \in W, w <_P v, M, v \models \varphi \\
M, w \models \Box_N\, \varphi &\iff \forall v \in W, v \leq_N w, M, v \models \varphi \\
M, w \models \overleftarrow{\Box}_N\, \varphi &\iff \forall v \in W, w <_N v, M, v \models \varphi
\end{aligned}$$

We can define the other operators $(\vee, \rightarrow, \Diamond, \overleftarrow{\Diamond})$ as usual. Finally, we can talk about a formula being true in all worlds or some worlds: $\overleftrightarrow{\Box}_P\, \varphi \equiv \Box_P\, \varphi \wedge \overleftarrow{\Box}_P\, \varphi$ and $\overleftrightarrow{\Diamond}_P\, \varphi \equiv \Diamond_P\, \varphi \vee \overleftarrow{\Diamond}_P\, \varphi$, respectively (similarly for normality). The following abbreviations are defined [4]:

(1) $I(\psi \mid \varphi) \equiv \overleftrightarrow{\Box}_P \neg\varphi \vee \overleftrightarrow{\Diamond}_P(\varphi \wedge \Box_P(\varphi \rightarrow \psi))$ (Conditional preference)

(2) $\varphi \leq_P \psi \equiv \overleftrightarrow{\Box}_P(\psi \rightarrow \Diamond_P \varphi)$ (Relative preference)

(3) $T(\psi \mid \varphi) \equiv \neg I(\neg\psi \mid \varphi)$ (Conditional tolerance)

(4) $\varphi \Rightarrow \psi \equiv \overleftrightarrow{\Box}_N \neg\varphi \vee \overleftrightarrow{\Diamond}_N(\varphi \wedge \Box_N(\varphi \rightarrow \psi))$ (Normative conditional)

The abbreviations state that (1) ψ is ideally true if φ is true, (2) φ is at least as preferred as ψ, (3) ψ is tolerable given φ and (4) that ψ normally is the case when φ is.

[1] We adopt the notion by Boutilier and others that we prefer minimal models, so $v \leq_P w$ denotes that v is at least as preferred as w.

In order to make decisions as motivated above, we define the following abbreviations, which allow us to specify different kinds of relative preference, and relative tolerance.

$$\varphi \not\leq_P \psi \equiv \neg(\varphi \leq_P \psi) \qquad \text{(Not as preferred)}$$

$$\begin{aligned}\varphi \approx_P \psi \equiv\ & (\varphi \leq_P \psi \wedge \psi \leq_P \varphi) \\ & \vee (\varphi \not\leq_P \psi \wedge \psi \not\leq_P \varphi)\end{aligned} \qquad \text{(Equally preferred)}$$

$$\begin{aligned}\varphi \leq_{T(\gamma)} \psi \equiv\ & (T(\varphi \mid \gamma) \wedge \neg T(\psi \mid \gamma)) \vee \\ & ((T(\varphi \mid \gamma) \leftrightarrow T(\psi \mid \gamma)) \wedge \\ & (\varphi \leq_P \psi \vee \varphi \approx_P \psi))\end{aligned} \qquad \text{(Relative tolerance)}$$

Relative tolerance is defined as φ being at least as tolerable as ψ w.r.t γ when either φ is tolerable given γ and ψ is not, or both φ and ψ are tolerable given γ (or both are not), and φ is at least as preferred as ψ, or they are equally preferable. This means that even if neither is tolerable, they are still comparable.

3.1 Making a Decision

We now show how QDT can be used to decide between conflicting influences. We define a model for an agent's decision making as follows:

$$\mathcal{M}_C = \langle M, F, C, B \rangle,$$

where M is a QDT-model as defined above, F is the set of influences, C is the set of controllable propositions[2], and B is the agent's belief base.

The set of potential consequences C' is defined such that if $\varphi \in C$ then $\varphi, \neg\varphi \in C'$. That is, if φ is controllable, then one of $\varphi, \neg\varphi$ may be a consequence of bringing about some state.

In order for a potential consequence to be an actual (expected) consequence of φ, it has to follow from the most normal worlds where φ holds. That is, we add φ to the belief base B, and the potential consequences that follow from the expanded belief base are then the expected consequences. Assuming that φ and B are consistent, we add φ to B using the *expansion* operator, $+$, of the AGM theory [1], where $B + \varphi$ means adding φ to a copy of B and closing the resulting set under logical consequence. We work with a copy of the belief base since the reasoning concerns what happens *if* the literal is added.

If, however, φ and B are not consistent, we can use the AGM *revision* operator, $\dot{+}$, which behaves like $+$, but if φ and B are not consistent, B is minimally modified to make it consistent with φ, before adding φ.

As shown in [3], AGM belief revision can be efficiently implemented in rational agents, making it suitable for our approach. We can now formally define the expected consequence of bringing about a state.

[2] A controllable proposition is, roughly, a proposition that the agent is able to influence, directly or indirectly, by an action. E.g., *snow* is not controllable and cannot be a consequence of an action, whereas *work* is.

Definition 1 (Expected consequences). *Given an agent's belief base B, the set of potential consequences C' and a literal φ. The expected consequences of bringing about φ, denoted $EC(\varphi)$, is given by:*

$$EC(\varphi) = \bigwedge C_\varphi \textit{ for all } C_\varphi \in \{C_\varphi \mid (B \dot{+} \varphi \Rightarrow C_\varphi) \textit{ where } C_\varphi \in C'\}$$

i.e. the conjunction of all literals C_φ that are normally consequences of the current belief base B expanded with φ, such that B remains consistent. If there are no expected consequences, then $EC(\varphi) = \top$.

Consider a normality ordering in which we have that

$$a \wedge x \Rightarrow b, \qquad a \wedge \neg x \Rightarrow c, \quad d \wedge \neg x \Rightarrow e,$$

and belief base $B = \{x\}$. Then we have that $EC(a) = b$ and $EC(d) = \top$. If $B = \{\neg x\}$, then $EC(a) = c$ and $EC(d) = e$.

Definition 2 (Most preferred influences). *Given an agent's set of influences F, the most preferred influences then are defined as the set Pref:*

$$\textit{Pref} = \{\varphi \mid \varphi \in F \wedge \forall \psi \in F \ (\psi \neq \varphi \rightarrow \varphi \leq_P \psi)\}$$

Definition 3 (Most tolerable consequences). *Given an agent's set of influences F, the most preferred influences then are defined as the set Tol:*

$$\textit{Tol} = \{\varphi \mid \varphi \in F \wedge \forall \psi \in F \ (\psi \neq \varphi \rightarrow EC(\varphi) \leq_{T(\varphi \vee \psi)} EC(\psi))\}$$

An agent can make a decision by selecting the most preferred influences having the most tolerable consequences from the set of potentially conflicting influences, F,.

Definition 4 (Decision). *Given a the set of influences F and the expected consequences $EC(\varphi)$ for all $\varphi \in F$, we can get the set of best influences (the decision) the agent should choose from, Dec, as follows:*

$$\textit{Dec} = \begin{cases} \textit{Tol} & \text{if } \textit{Tol} \cap \textit{Pref} = \emptyset \\ \textit{Tol} \cap \textit{Pref} & \text{otherwise} \end{cases}$$

Given a model $\mathcal{M}_C$, an agent can then choose an arbitrary literal from *Dec*, since all of these will be preferred and have tolerable consequences (or at least have tolerable consequences).

If there are no expected consequences of bringing about a certain proposition, i.e. if $EC(\varphi) = \top$, then φ is considered tolerable since we do not expect any consequences. Therefore comparing the relative tolerance for all other consequences, ψ, is reduced to comparing $\top \leq_{T(C)} \psi$ and $\psi \leq_{T(C)} \top$. Note that $T(\top \mid \psi)$ is true iff ψ is true in any world[3]. Furthermore, $\top \leq_P \psi$ is always true, and $\psi \leq_P \top$ is true iff ψ is true in all worlds. Thus, it is possible to make a decision even if some influences have no known consequences.

[3] Since $T(\top \mid \psi) \equiv \overleftrightarrow{\Diamond}_P \psi \wedge \overleftrightarrow{\Box}_P(\neg\psi \vee \Diamond_P(\psi \wedge \top))$.

In the following, we show that an agent given a model, $\mathcal{M}_C$ can always make a decision.

Lemma 1. *Given expressions φ, ψ, and γ, the following relation holds for relative tolerance:*

$$\neg(\varphi \leq_{T(\gamma)} \psi) \rightarrow (\psi \leq_{T(\gamma)} \varphi)$$

Proof. We assume $\neg(\varphi \leq_{T(\gamma)} \psi)$ and prove that $(\psi \leq_{T(\gamma)} \varphi)$. Based on the assumption and the definition of relative tolerance, the following formulas hold:

$$\neg(T(\varphi \mid \gamma) \wedge \neg T(\psi \mid \gamma)) \tag{5}$$

$$\neg((T(\varphi \mid \gamma) \leftrightarrow T(\psi \mid \gamma)) \;\wedge\; (\varphi \leq_P \psi \vee \varphi \approx_P \psi)) \tag{6}$$

1. Given (5), we have that either $T(\varphi \mid \gamma) \leftrightarrow T(\psi \mid \gamma)$ or $\neg T(\varphi \mid \gamma) \wedge T(\psi \mid \gamma)$ holds. In the latter case we have that $\psi \leq_{T(\gamma)} \varphi$ by the definition of relative tolerance. Otherwise they are equally tolerable and we have to consider the second case.
2. Given (6), either $\neg(T(\varphi \mid \gamma) \leftrightarrow T(\psi \mid \gamma))$ or $\neg(\varphi \leq_P \psi \vee \varphi \approx_P \psi)$. If the former is the case, then one is tolerated and the other is not. Because of (5), we have that $\neg T(\varphi \mid \gamma) \wedge T(\psi \mid \gamma)$ and therefore $\psi \leq_{T(\gamma)} \varphi$. If the latter is the case then we have that $\neg(\varphi \leq_P \psi) \wedge \neg(\varphi \approx_P \psi)$. In that case we have that $\psi <_P \varphi$ and therefore $\psi \leq_{T(\gamma)} \varphi$. □

Proposition 1. *Given a non-empty set of influences F and the expected consequence $EC(\varphi)$ for each $\varphi \in F$, the set of best influences, Dec, is always non-empty.*

Proof. If $|F| = 1$ then $Dec = F = Tol = Pref$, since there are no $\psi \neq \varphi$ in F. If $|F| > 1$ then we consider each case.

- If $Tol \cap Pref = \emptyset$ then $Dec = Tol$ and we have to show that $Tol \neq \emptyset$. If $Tol = \emptyset$ then there is no ψ such that $EC(\varphi)$ is relatively more tolerated than $EC(\psi)$. Since $|F| > 1$ there is at least one $\psi \neq \varphi$, and by lemma 1 we then have that $EC(\psi)$ is relatively more tolerated than $EC(\varphi)$. Thus $\psi \in Tol$ and $Tol \neq \emptyset$.
- If $Tol \cap Pref \neq \emptyset$ then, since $Dec = Tol \cap Pref$, Dec cannot be empty. □

Proposition 1 shows that the decision procedure will always produce a non-empty result, meaning that we can use the procedure even in situations where there is no conflict between influences.

4 Generating Models

The preferences of an agent are usually not described as a model shown above, but will rather be expressions such as “I prefer that it does not rain” or “When it

Algorithm 1. Atom retrieval

```
function RETRIEVE_ATOMS(F, R)
    At ← POSITIVE(F)
    checked ← ∅
    for all φ ∈ At \ checked do
        At ← At ∪ ATOMS_RULE(φ, R)
        checked ← checked ∪ {φ}
    return At
```

rains, I want to stay inside". In order to utilize such preferences in the decision procedure above, a transformation is required. In the following, we present a method, which will generate a QDT-model that respects non-contradictory rules specified by the agent.

Each agent specifies a set of rules of the form (φ, ψ), where φ and ψ are standard propositional formulas. A rule, (φ, ψ), should be read as "if φ then normally/preferably ψ". Using the notion of possible worlds, we understand a rule as follows. Worlds w, in which $w \models \varphi \wedge \psi$, are favored over worlds w', where $w' \models \varphi \wedge \neg\psi$. Thus, a rule is roughly interpreted as the conditionals for preference and normality. In the following, we propose a method for generating preference and normality orderings that respect such rules by utilizing this interpretation. The generic definition of the conditional operators from the previous section is:

$$\textit{if } \varphi \textit{ then } \psi \equiv \overleftrightarrow{\Box}\neg\varphi \vee \overleftrightarrow{\Diamond}(\varphi \wedge \Box(\varphi \rightarrow \psi)).$$

From this definition, it is clear that there are two ways to ensure that a rule (φ, ψ) is respected. Either (a) φ is never true or (b) in the most favored world(s) where φ is true, ψ is also true. Option (a) is achieved easily; we simply remove all worlds where φ is true. However, the agent does probably not intend this, since the rules are most likely specified such that favored situations are actually also possible situations. We therefore require that the method does not remove any worlds from W. The method should ensure that after the application of a rule we have $M \models (\varphi, \psi)$. Another natural requirement is that previously applied rules still hold after application of a new rule. If this is not possible, we say that the new rule contradicts previously applied rule, and therefore discard the new rule.

The aim is to generate a model respecting the rules, such that the agent can make a decision based on the model. Given the modal nature of QDT, the generation is based on the notion of possible worlds, W, so the first step is to generate W. Instead of generating a general model in which all rules are applicable, we create sub-models for different parts of the world. For example, an agent's preference concerning work might not be relevant for decisions in a different context, such as a family party. Furthermore, certain situations are not deemed possible, such as leaving work early and not going to work at all. W is generated using the agent's current influences F to decide which atoms

Algorithm 2. Rule application

function APPLY$((\varphi, \psi), W, \leq)$
  $max \leftarrow max(\leq)$
  for all $w \in W$ **do**
    if $w \models \varphi \wedge \neg\psi$ **then** $W_c \leftarrow w$
    if $(w \models \varphi \wedge \psi)$ and $\neg\exists w'(w' \in W \wedge (w', w) \in lock)$ **then**
      $o(w) = max + 1$
      $W_s \leftarrow w$
  if $W_s = \emptyset$ **then return** $\perp$
  for all $w \in W_s, w' \in W_c$ **do** $lock(w, w')$
  return $\top$

are relevant for making a decision in the context of F and any impossible worlds are removed. Algorithm 1 retrieves the relevant atoms from F and the set of rules. POSITIVE(S) is the set of all literals in S with all negative literals made positive, such that if $\neg\varphi \in S$ then $\varphi \in$ POSITIVE(S). ATOMS_RULE$(\varphi, \mathcal{R})$ returns a set of all atoms that appear in rules $r \in \mathcal{R}$ where φ also appears (e.g. if $\mathcal{R} = \{(\varphi, \psi_1), (\varphi, \psi_2)\}$ then ATOMS_RULE$(\varphi, \mathcal{R}) = \{\varphi, \psi_1, \psi_2\}$.

Given the set of relevant atoms, At, the set of possible worlds contains a world for each set in 2^{At}, where each set either contains the atom or its negation. For instance, given $At = \{a, b\}$, the initial model will be $2^{At} = \{\{a, b\}, \{\neg a, b\}, \{a, \neg b\}, \{\neg a, \neg b\}\}$. Impossible worlds are specified as simple formulas, e.g. $\neg a \wedge b$. A world that entails such an expression is removed from W, which is then set of possible worlds given F.

An ordering, $\leq$, is the result of a mapping from a world to a natural number, the o-value, denoted $o : W \to \mathcal{N}$, such that worlds with higher numbers are more favored. Worlds can have the same o-value if they are equally favored. The maximum o-value of an ordering $\leq$ is denoted $max(\leq)$.

We propose using a *locking* mechanism in which the ordering between two worlds can be locked, such that if $lock(w_1, w_2)$ then it must always be the case that $w_1 < w_2$. We can use this to e.g. lock the ordering between worlds $w_1 = \{\varphi, \psi\}$ and $w_2 = \{\varphi, \neg\psi\}$ if a rule (φ, ψ) is applied by creating a lock, $lock(w_1, w_2)$, such that w_1 is always favored over w_2. Then if a rule $(\varphi, \neg\psi)$ is applied, the ordering cannot be changed so that w_2 is favored over w_1 because it would result in the previously applied rule no longer being respected (since ψ would not be entailed by the most favored world where φ holds).

Rules are applied using the function APPLY : $(\mathcal{R}, \leq) \to \{\top, \perp\}$ (algorithm 2). Applying a rule (φ, ψ) is done by finding all worlds in which both φ and ψ holds (the sought worlds) and all worlds in which φ and $\neg\psi$ holds (the contradictory worlds). The sought worlds are given an o-value of $max(\leq) + 1$ and all contradictory worlds are locked in relative position to the sought worlds.

A rule (φ, ψ) cannot be applied if there is no world w in which $w \models \varphi \wedge \psi$ or for all such worlds a lock, $lock(w', w)$, exists for some w'.

Proposition 2. *Given an initial ordering $\leq$ and a set of rules $\mathcal{R} = \{r_1, \ldots, r_n\}$ where each r_i is of the form (φ_i, ψ_i), the result of successfully applying rules r_1 to r_i, $0 < i \leq n$ is an ordering which respects rules $\{r_1, \ldots, r_i\}$.*

Proof. When $i = 1$ no previous rules have been applied, so we only have to show that the model respects rule r_1 after successful application. We have $o(w) = 1$ for all worlds w. Applying r_1 can only fail if no worlds entail $\varphi_1 \wedge \psi_1$ or all entailing worlds are locked. Since $lock = \emptyset$ initially, only the former can be the case. But then the rule would describe an impossible world and cannot be applied. Otherwise, after applying r_1, it is entailed by the model, since for all worlds w where $w \models \varphi_1 \wedge \psi_1$ we have $o(w) = 2$ and the o-value of all other worlds is unchanged. Thus the worlds entailing r_1 are most preferred so the rule itself is entailed by the model.

When $i > 1$ we assume that all rules up to and including r_{i-1} have been applied successfully. We therefore have

$$M \models (\varphi_1, \psi_1) \wedge \cdots \wedge (\varphi_{i-1}, \psi_{i-1}).$$

Let $l_i = \{(w, w') \mid w \models \varphi_i \wedge \psi_i \text{ and } w' \models \varphi_i \wedge \neg\psi_i\}$ be the set of locks between worlds with contradictory consequents of a rule (φ_i, ψ_i). Before applying r_i the set $lock$ contains

$$lock = l_1 \cup \cdots \cup l_{i-1}$$

Rule r_i can then be applied if there is at least one world w in which $w \models r_i$ and where w is not the second entry of a pair in $lock$ (i.e. there is a world entailing r_i which is not locked by another world). If there is no such world then either the rule describes an impossible world and should be rejected, or a previously applied rule contradicts it, which also means it should be rejected. Otherwise the rule will be successfully applied resulting in a model entailing all rules up to and including r_i:

$$M \models (\varphi_1, \psi_1) \wedge \cdots \wedge (\varphi_i, \psi_i),$$

and a new $lock$ set: $lock' = lock \ \cup \ l_i$. Assuming that the rule is successfully applied we know that for all w in which $w \models r_i$ we have $o(w) = max(\leq) + 1$. Clearly r_i is then entailed by the model. We then have to show that all rules up to r_i are still entailed as well.

Consider rule r_j where $0 < j < i$. Rule r_j was entailed by the model before applying r_i. Therefore there are worlds w_j where $w_j \models \varphi_j \wedge \psi_j$ and no lock of it exists, and w'_j where $w'_j \models \varphi_j \wedge \neg\psi_j$, and for all such worlds we have that $o(w_j) > o(w'_j)$ and $(w_j, w'_j) \in lock$. Thus all worlds contradicting r_j are locked relative to those entailing it. If $w'_j \in W_s$ for some w'_j then some of the sought worlds are locked by r_j, but since W_s only contains unlocked worlds, this cannot be the case. Therefore no worlds w'_j will be given a higher o-value than any w_j world. Furthermore, since w'_j contains all the worlds that could invalidate r_j, clearly r_j is still entailed after applying r_i. □

Algorithm 3. Model generation

```
function GENERATE(F, P, R)
    At ← RETRIEVE_ATOMS(F, R)
    W ← INIT(At, P)
    ≤ ← o(W)
    R' ← SORT(R)
    for all (φ, ψ) ∈ R' do
        APPLY((φ, ψ), W, ≤)
    return ≤
```

Even though we can successfully apply a set of rules, the function can be further optimized to maximize the number of successful applications of rules. Note that the use of a locking mechanism decreases the number of worlds that can be moved around every time a rule is successfully applied. Therefore, by minimizing the number of worlds being locked in each iteration, we maximize the number of rules that can be applied. The function $s : \mathcal{R} \rightarrow \mathcal{N}$ gives each rule a score, where rules with many propositions and operators receive higher scores than rules with few.

$$\begin{aligned}
s((\top, \psi)) &= s(\psi) - 1 \\
s((\varphi, \psi)) &= s(\varphi) + s(\psi) \\
s(\varphi \wedge \psi) &= s(\varphi) + s(\psi) + 1 \\
s(\varphi \vee \psi) &= s(\varphi) + s(\psi) + 1 \\
s(\neg\varphi) &= s(\varphi) + 1 \\
s(\top) &= 0 \\
s(p) &= 1
\end{aligned}$$

By applying the highest valued rules (the most specialized) first, we ensure that as few worlds as possible are locked. Notice that that rules where the antecedent is $\top$ will be penalized, since they are very general, whereas $\top$ in the consequent is ignored.

The algorithm GENERATE : $(F, \mathcal{P}, \mathcal{R}) \rightarrow \leq$ then works as follows (algorithm 3). Retrieve relevant atoms and generate an initial model of possible worlds. Sort rules descending according to their s-value using SORT($\mathcal{R}$). Each rule in $\mathcal{R}$ is then applied using APPLY($(\varphi, \psi), \leq$). Finally, the algorithm returns the ordering $\leq$, which respects all successfully applied rules.

4.1 Application of Equally General Rules

The need for constraining the order of rule application touches upon a shortcoming of the model generation; rule application may fail, if previously applied rules have locked the matching worlds. In many cases this is actually a good thing, since it does not make sense to first apply a rule $r_1 = (\varphi, \psi)$ and then later $r_2 = (\varphi, \neg\psi)$. r_1 and r_2 are clearly contradictory rules, and both should not be applied at once, since we cannot both expect ψ and $\neg\psi$ when φ is true. However, if two rules receive the same score they will be applied in a non-deterministic order which could lead to a situation where applying the rules in one order results in one model, and applying in another order results in a different model. It might even be the case that we can apply both rules using one ordering, while another ordering rejects one of the rules.

Consider rules $\mathcal{R} = \{(\top, A), (\top, B)\}$ and possible worlds $W = 2^{\{A,B\}}$. The rules have equal score and they will therefore be applied in a non-deterministic order. If $(\top, A)$ is applied first the ordering will be $AB < A\overline{B} < \{\overline{A}B, \overline{AB}\}$, whereas the ordering will be $AB < \overline{A}B < \{A\overline{B}, \overline{AB}\}$ if $(\top, B)$ is applied first. The rules satisfy the model in both cases; in the most preferred world(s) both A and B hold, but the ordering of less preferred world differs. We argue that even though this is the case, it is clear that as long as the rules have been successfully applied they are satisfied by the model, which means that the model can be used by the agent to reason about its influences by taking its preferences into account. In situations where certain orderings might reject a rule while other orderings would not, it is evident that the latter ordering is favored[4]. If this is the case, the agent might simply monitor the rule application, and if the algorithm rejects a rule given a certain ordering, the agent can attempt to apply the equally general rules in a different order. However, if all orderings result in rejection of one of the rules, this indicates that some of the rules contradict each other, suggesting that not all the rules can be consistently applied to the model.

5 Case Study

In this section, we apply the model to a simple scenario. We consider a situation in which agents are normally expected to go to work, but during snowy weather, they are not expected to go to work. The agent Alice prefers that it does not snow, but when it snows, she wants to stay at home. We have the following rules for expectations of the environment and preferences of the agent:

$$\mathcal{R}_{Env} = \{(\top, work), (snow, \neg work)\}$$
$$\mathcal{R}_{Alice} = \{(\top, \neg snow), (snow, \neg work)\}.$$

The environment expectation rules represent the expectations that originate from different sources such as an organization or other agents.

In the following we let S abbreviate *snow* and W *work*. We denote negation using an overline, e.g. $\overline{S}$ when it is not snowing and we write conjunctions by writing literals next to each other, e.g. SW when it is snowing and the agent is working. From the rules above it is clear that $At = \{W, S\}$. The orderings $\leq_P$ and $\leq_N$ are then generated using the algorithms described above. Figure 1 shows how Alice's preference ordering is generated using her rules.

To make the situation more interesting we add the possibility of *being fired* (F) and of *leaving early* (E):

$$\mathcal{R}_{Alice} = \{(\top, \overline{S}), (S, \overline{W}), (\top, \overline{F}), (W, E)\}.$$

[4] After all, the aim is to apply as many of the rules as possible.

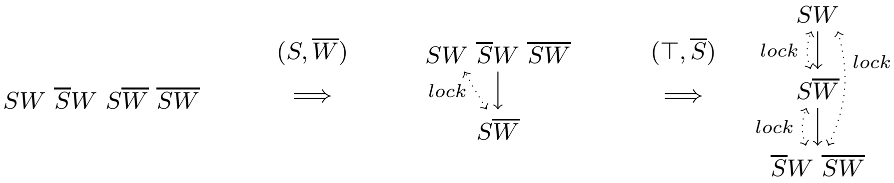
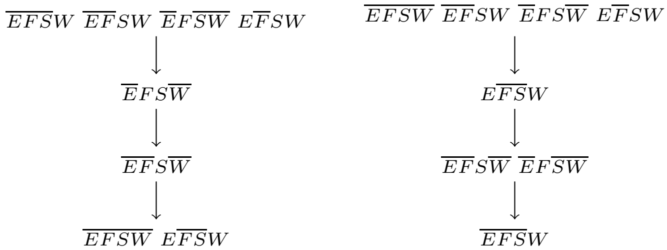


Fig. 1. Generation of Alice's preferences. Note that some of the locks have been omitted for clarity, e.g. the lock between $\overline{S}W$ and SW.



(a) Alice's preferences (b) Expectation

Fig. 2. The preference and normality orderings generated using the rules and prohibitions specified by the environment and Alice

Thus, she does not want to be fired, and in situations where she chooses to go to work, she prefers to leave early. The rules of the environment are updated to conform to this change; if it snows, one can stay home without being fired but this is not the case when it does not snow.

$$\mathcal{R}_{Env} = \{(\top, W), (S, \overline{FW}), (\overline{SW}, F), (\top, \overline{E}), (W, \overline{F})\}.$$

Furthermore, agents are not expected to leave early and will normally not be fired if they work.

Certain worlds are not possible given the new rules; an agent will not be working if it is fired, and if it is not working, it will not leave early. This is represented by the set of prohibitions: $\mathcal{P} = \{FW, E\overline{W}\}$. Thus, the set of possible worlds W is reduced to those worlds where none of the prohibitions above are entailed. The preference and normality orderings resulting from these rules are shown in figure 2(a) and 2(b).

Alice is now able to decide between her influences using the generated model. Say Alice has a desire to stay at home, but an obligation toward her employer to go to work, i.e. the set of influences is $F = \{W, \overline{W}\}$. We then consider two cases: one where it snows and one where it does not.

a) We have that $B = \{S\}$ so all worlds in which it does not snow are ignored. This leaves us with four possible worlds, where Alice's most preferred world is $\overline{EF}S\overline{W}$, thus $Pref = \{\overline{W}\}$. The expected consequence of both going to work and not going to work is not to be fired, which means that each is equally tolerable, thus $Tol = \{W, \overline{W}\}$. The decision is then the intersection: $Dec = \{\overline{W}\}$.

b) We have that $B = \{\overline{S}\}$, giving four possible worlds. In this case Alice's most preferred worlds are $\overline{E}\overline{F}\overline{S}\overline{W}$ and $E\overline{F}\overline{S}W$, thus $Pref = \{W, \overline{W}\}$. From the expectations we see that $EC(W) = \overline{E}\overline{F}\overline{S}$ and $EC(\overline{W}) = \overline{E}F\overline{S}$. Since not being fired is more tolerable than being fired, $Tol = \{W\}$, and the decision is then $Dec = \{W\}$.

Note that Alice was labeled neither "social" nor "selfish". Her preference and the expected consequences are taking into account, and this leads to the results above. When she chooses to go to work, this does not mean that she is strictly social. She might very well have a (selfish) desire to leave early, which she can choose to do if she tolerates the consequences of doing so.

6 Toward an Implementation

The case study showed that agents are able to make decisions based on rules of preference and expectation. We believe that the approach can be integrated in existing agent systems to let agents make decisions based on their own preferences and the external expectations. We are currently investigating how the procedure can be integrated into the GOAL agent programming language [9]. While this is work in progress, we briefly discuss the work that has been done and some of the implications such integration has.

In GOAL, the choice of committing to different goals and performing actions is relatively simple; a program consists of a list of rules that are either evaluated in linear or random order. This means that either the preference ordering is specified a priori, or it is not specified at all. We believe that by integrating the agents' rules of preference and the expectations into the GOAL system, the agents will be able to make decisions based on preferences in different situations thus providing a different kind of processing order of GOAL rules. This requires that the system is able to understand a specification of preferences and expectations.

We have taken the first steps toward an implementation by implementing a prototype of the system in Prolog[5]. The reason for choosing Prolog is that (1) it makes the implementation of the QDT models quite simple and (2) it allows us to integrate the system directly into the GOAL agent's knowledge base. The set of rules is specified as a list of pairs, `[(Phi,Psi),...]`; prohibitions as simple formulas; and a lock as a pair of lists, such that `(L1,L2)` represents that for all worlds w_1 in `L1` and w_2 in `L2` is it the case that $w_1 < w_2$.

The basic operators $(\wedge, \neg, \Box)$ are implemented straightforwardly; $\wedge$ and $\neg$ are evaluated in the current world and $\Box$ in all more preferred (or expected) worlds.

[5] The Prolog code that follows has been slightly simplified to be more easily comprehended.

Each abbreviation is then defined, e.g. the conditional preference operator is defined as follows:

```
eval(I(ψ | φ), Ws, W, TV) :-
    eval(□⃡_P¬φ ∨ ◇⃡_P(φ ∧ □_P(φ → ψ)), Ws, W, TV).
```

where Ws is the set of all worlds and W is the current world. eval succeeds if TV can be unified with the truth-value of the formula.

The application of a rule is done using two findall-queries: one to build the set W_c and one for W_s.

```
apply_rule(Ws, Ord, (φ,ψ), Lock, W_c, W_s) :-
    findall(W, (member(W,Ws), eval(φ∧¬ψ, Ws, W, t)), W_c),
    findall(W, (member(W,Ws), \+ (member((_,Locked), Lock),
member(W, Locked)), eval(φ∧ψ, Ws, W, t)), W_s).
```

where Ws is the set of all worlds, Ord is the current ordering, (φ, ψ) is the rule being applied, Lock is the set of locks, and W_c and W_s are W_c and W_s, respectively. The first query succeeds if W_c can be unified with all worlds w in which $w \models \varphi \wedge \neg\psi$. The second query succeeds if W_s can be unified with all worlds w where $w \models \varphi \wedge \psi$ *and* w is not locked. A rule is successfully applied when W_s \= [], i.e. $W_s \neq \emptyset$. The ordering can be changed by incrementing the o-value for each $w \in W_s$, and the lock is updated to include the pair of lists (W_s, W_c).

Agents make a decision using the sets *Pref* and *Tol*, which are built by following their definitions closely. For example, the set *Pref* is built as follows:

```
pref([],_,_,[]).
pref([φ|FTail], F, Ws, Pref) :-
    checkpref(φ, F, Ws), !, Pref=[φ|Tail],
    pref(FTail, F, Ws, Tail).
pref([_|FTail], F, Ws, Pref) :- pref(FTail, F, Ws, Pref).
```

where F is the set of all influences, Ws is the set of all worlds, and checkpref(Phi, F, Ws) succeeds if $\varphi \leq_P \psi$ for all $\psi \in F$. Pref is then unified with all $\varphi \in F$ that are most preferred. A similar predicate is defined for *Tol*. The final set, *Dec*, is the intersection of *Pref* and *Tol*, or just *Tol* if the intersection is empty, and a decision can then be made using the following Prolog query (here making a decision based on the case study above):

```
?- decision([¬s], P, N, Dec).
Dec = [w].
```

where P and N are the generated preference and normality orderings, and Dec corresponds to *Dec*.

The decision procedure can be used as-is within GOAL, meaning that GOAL agents are able use the decision procedure. However, this also means that the decision of which influence to commit to needs to be implemented directly in the agent's program, which suggests that the programmer will have to understand the mechanisms of the procedure. A more ideal solution would be to integrate the procedure within GOAL, e.g. allowing for another GOAL rule evaluation order (which would then choose a rule matching an influence in *Dec*), requiring only that the programmer to specifies each agent's preferences and the expectations from the environment. This is however out of scope for this paper and is left for future research.

7 Conclusion

We have argued that conflicts are prone to arise when agents interact in open societies and enact roles in an organization, since their own desires may be in conflict with obligations toward other agents or the obligations of the role(s) they are enacting. We have discussed why obligations along with desires should be considered influences on the agent's behavior rather than being seen as desires being imposed onto the agent by other entities. Since influences do not necessarily represent states the agent wants to achieve, they should only be pursued if the agent can tolerate their consequences.

Our approach to decide which influences to commit to, which is based on qualitative decision theory, is an attempt to let the agent reason about the influences without taking into account that one influence is a desire, and another is an obligation, since such bias can result in labeling the agent "selfish" or "social" in advance. The approach works by including the consequence of bringing about a state in the reasoning, thus letting the agent consider its preferences, without choosing something that results in an intolerable state. We have argued that this indeed lets the agents reach a decision without strictly preferring certain types of influences to others.

To make the procedure readily available we furthermore have developed a simple method that can generate models to be used in the reasoning process by the use of expressions describing the agent's preferences. By use of a simple locking mechanism, the method generates models, which respect non-contradictory rules specified by the agent such that it is possible to make a decision among a set of influences. The simple nature of the method also allows us to generate the models on the fly, so that if the agent's preferences change during execution a new model can be generated. Since the method works by generating all possible states of relevant sub-models, it may prove to be inefficient in cases that are more complex. Even though we only consider sub-models, it would be natural to investigate how to optimize this. Furthermore, since rules of equal generality are applied non-deterministically, different models may emerge, though they satisfy the same sets of rules; although our goal was to create models satisfying rules, we believe a deterministic procedure is desirable and it could be an interesting direction for future work.

Another direction for further research would be to investigate how to integrate the prototype into the GOAL agent programming language. While we have already built a working prototype of the system in Prolog, much more work needs to be done to successfully integrate it into a full-fledged programming language such as GOAL.

Finally, the non-propositional case should be investigated such that reasoning about the agent's preferences can be done in cases that are more complex. For instance, it should be possible for the agent to prefer being at home, *at(home)*, compared to other places such as work, while still being able to express that being at a café is more preferred than being at home.

Acknowledgement. The author would like to thank the reviewers for their useful comments and the attendants at the EMAS workshop for the interesting discussion.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. The Journal of Symbolic Logic 50(2), 510–530 (1985)
2. Alechina, N., Dastani, M., Logan, B.: Programming Norm-Aware Agents. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, vol. 2, pp. 1057–1064. International Foundation for Autonomous Agents and Multiagent Systems, Richland (2012)
3. Alechina, N., Jago, M., Logan, B.: Resource-bounded belief revision and contraction. In: Baldoni, M., Endriss, U., Omicini, A., Torroni, P. (eds.) DALT 2005. LNCS (LNAI), vol. 3904, pp. 141–154. Springer, Heidelberg (2006)
4. Boutilier, C.: Toward a Logic for Qualitative Decision Theory. In: Proceedings of the KR 1994, pp. 75–86. Morgan Kaufmann (1994)
5. Broersen, J., Dastani, M., Hulstijn, J., Huang, Z., Torre, L.v.d.: The BOID Architecture – Conflicts Between Beliefs, Obligations, Intentions and Desires. In: Proceedings of the Fifth International Conference on Autonomous Agents, pp. 9–16. ACM Press (2001)
6. Dastani, M., Dignum, V., Dignum, F.: Role-assignment in open agent societies. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2003, pp. 489–496. ACM, New York (2003)
7. Dignum, F., Morley, D., Sonenberg, E.A., Cavedon, L.: Towards Socially Sophisticated BDI Agents. In: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS 2000), pp. 111–118. IEEE Computer Society, Washington, DC (2000)
8. Dignum, F., Kinny, D., Sonenberg, L.: From Desires, Obligations and Norms to Goals. Cognitive Science Quarterly 2 (2002)
9. Hindriks, K.V.: Programming Rational Agents in GOAL. Multi-Agent Programming: Languages, Tools and Applications 2, 119–157 (2009)
10. van der Torre, L., Tan, Y.H.: Contrary-to-duty reasoning with preference-based dyadic obligations. Annals of Mathematics and Artificial Intelligence 27(1-4), 49–78 (1999)