Behaving Responsible in Multi-Agent Worlds

(Extended Abstract)

Tiago de Lima
Eindhoven University of Technology
P.O. Box 513, 5600MB
Eindhoven, The Netherlands
t.d.lima@tue.nl

Lambèr Royakkers
Eindhoven University of Technology
P.O. Box 513, 5600MB
Eindhoven, The Netherlands
l.m.m@royakkers.nl

Frank Dignum
Utrecht University
P.O. Box 80089, 3508TB
Utrecht, The Netherlands
dignum@cc.uu.nl

ABSTRACT

It has been proposed that a good way of allocating tasks to agents is by ascribing them obligations, i.e., if we want agent i achieves φ, we can stipulate that 'it is obligatory for i that φ'. Here, we argue that this method is not adequate to guide agent's decisions. Then, using a multi-agent extension of propositional dynamic logic, with operators expressing agents' knowledge and abilities, we show that when agents' decisions are guided by responsibilities, as we define here, a successful performance is more likely to be obtained.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: multi-agent systems; I.2.4 [**Knowledge Representation Formalism and Methods**]: Modal Logic

General Terms

Theory

Keywords

Responsibility, obligations, deontic logic

1. INTRODUCTION AND MOTIVATION

A natural way of allocating tasks to agents is by ascribing them obligations. For example, if we want agent i achieves outcome φ, we can stipulate that 'it is obligatory for i that φ'. Then, agent i can decide how to behave, by inferring which actions are forbidden, permitted and obligatory. The latter can be done using the reduction from obligations-to-be to obligations-to-do, proposed in [5, 2].

Here we argue that the method mentioned above is not completely adequate to guide agent's decisions. We illustrate our point using an example.

Example 1. A company has two bank accounts, 1 and 2. Bob will pay a bill using account 1. Alice must keep the balances of the accounts non-negative, i.e., 'it is obligatory for Alice that accounts 1 and 2 are non-negative'. She knows that Bob will withdraw from one of these accounts but she does not know which one. None of the accounts have enough money to cover the payment, but the total amount in the two accounts together is enough to cover it.

Cite as: Behaving Responsible in Multi-Agent Worlds, (Extended Abstract), Tiago de Lima, Lambèr Royakkers, Frank Dignum, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1139–1140

This example can be modelled by the model in Fig. 1, where p means that both bank accounts are non-negative, v_a means that Alice is in violation (i.e, a bank account is negative), the arrows represent the joint actions Alice and Bob may execute, and the dashed lines represent Alice's knowledge. Note that Alice can fulfil her obligation by transferring some money from account 2 to 1 but, because her knowledge is incomplete, she also considers it possible that she should transfer from 1 to 2. Because Alice has the possibility to ask Bob before making the transfer, she can acquire the necessary information to decide what to do. Then, one may conclude that she must ask Bob before making the transfer.

However, the latter intuitive conclusion cannot be achieved using the idea proposed above. Alice has the possibility to acquire the necessary information but she is not able to decide to do so. Indeed, Alice is not obliged to ask Bob, because, if she does not do so, she will not be in a violation state. She will be in a violation state, if any, only after the second step, when Bob pays the bill.

In the next sections we propose a formalism to deal with situations as in Example 1. We build a logic that we call Coalition PDL (CPDL). It is an extension of PDL [3] with new elements expressing enacted actions, agents' knowledge and agents' abilities.

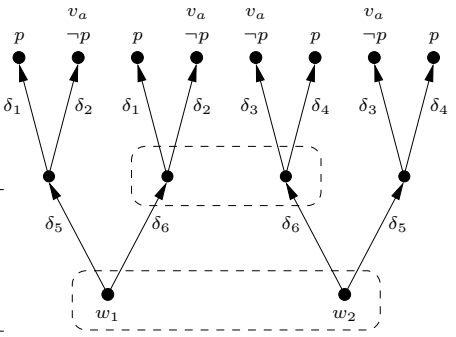
2. COALITION PDL

Assume a countable non-empty set P of atomic formulae, a finite non-empty set N of agents and a finite non-empty set A of atomic actions. We denote by Δ the set of all total functions $\delta : N \to A$ (the joint actions available for the agents). The language $\mathcal{L}$ of CPDL is defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathbf{K}_i\varphi \mid [G{:}\delta]\varphi \mid \langle\!\langle G\rangle\!\rangle\varphi$$

where $p \in P$, $G \subseteq N$, and $\delta \in \Delta$. The event $G{:}\delta$ means: 'the agents in G simultaneously execute their respective actions in δ (and we do not consider what the other agents are doing)'. The formula $[G{:}\delta]\varphi$ is read: 'φ holds after every possible execution of δ by the agents in G', and the formula $\langle\!\langle G\rangle\!\rangle\varphi$ is read: 'the group of agents G has the power to bring about φ'.

Formulae in $\mathcal{L}$ are interpreted in models consisting in tuples of the form $\langle W, \mathcal{K}, \mathcal{T}, \mathcal{V}\rangle$, where W is a non-empty set possible worlds; $\mathcal{K} : N \to (W \times W)$ defines, for each



	Alice	*Bob*
δ_1	*tr1*	*wth1*
δ_2	*tr2*	*wth1*
δ_3	*tr1*	*wth2*
δ_4	*tr2*	*wth2*
δ_5	*ask*	*wait*
δ_6	*wait*	*wait*

Figure 1: Model for Example 1

$i \in N$, an equivalence relation representing i's knowledge; $\mathcal{T} : \Delta \to (W \to W)$ defines, for each (δ, w), the state resulting from the performance of δ at w; and $\mathcal{V} : P \to 2^W$ defines the interpretation of atomic formulae. An example of such models is given in Fig. 1.

The satisfaction relation $\models$ is defined as usual for Boolean operators, and is the following for the modal operators:

$$\begin{array}{lll} M, w \models \mathbf{K}_i\varphi & \text{iff} & M, w' \models \varphi, \text{ for all } w' \in \mathcal{K}_i(w) \\ M, w \models [G{:}\delta]\varphi & \text{iff} & M, w' \models \varphi, \text{ for all } w' \in \mathcal{T}_{G:\delta}(w) \\ M, w \models \langle\!\langle G \rangle\!\rangle \varphi & \text{iff} & \text{there is } \delta \in \Delta \text{ such that} \\ & & \mathcal{T}_{G:\delta}(w) \neq \emptyset \text{ and} \\ & & M, w' \models \varphi \text{ for all } w' \in \mathcal{T}_{G:\delta}(w) \end{array}$$

with:

$$\begin{array}{l} \mathcal{T}_{G:\delta}(w) = \{w' \mid w' \in W \text{ and there is } \delta' \in \Delta \\ \qquad\qquad \text{s.t. } \delta'(i) = \delta(i), \text{ for all } i \in G, \\ \qquad\qquad \text{and } (\mathcal{T}(\delta')(w)) = w'\} \end{array}$$

We also assume that models satisfy the No-Forgeting constraint [4], thus being a bit different from the models in [8]. Operator [.] is similar to the one proposed in [7] and operator $\langle\!\langle . \rangle\!\rangle$ is similar to the one proposed in [1, 6].

Obligations are added by adapting the idea of [5, 2]. That is, we augment the set P by $\{v_G \mid G \in 2^N \setminus \emptyset\}$, and define the abbreviations:

$$\mathbf{O}_G\varphi \stackrel{\text{def}}{=} \langle\!\langle \emptyset \rangle\!\rangle (\neg\varphi \to v_G)$$
$$\mathbf{F}_G(\delta) \stackrel{\text{def}}{=} [G{:}\delta] v_G$$
$$\mathbf{P}_G(\delta) \stackrel{\text{def}}{=} \neg \mathbf{F}_G(\delta)$$

The formula $\mathbf{O}_G\varphi$ means 'it is obligatory for G that φ', and formulae $\mathbf{F}_G(\delta)$ and $\mathbf{P}_G(\delta)$ mean, respectively: 'action δ is forbidden for G' and 'action δ is permitted for G'.

3. RESPONSIBILITY

The following formula is true in w_1 in the model of Fig. 1:

$$\mathbf{K}_a(\langle\!\langle \emptyset \rangle\!\rangle (v_a \to \neg p) \wedge \mathbf{O}_a p) \wedge \neg \mathbf{K}_a \neg [a{:}\delta; \delta'] \neg p \wedge \neg \mathbf{K}_a \neg \mathbf{P}(\delta; \delta')$$

where $\delta(a)$ is either *ask* or *wait*, and $\delta'(a)$ is either *tr1* or *tr2*. Intuitively, this formula means that Alice knows her obligation, she knows that such sequence of actions possibly lead to a violation, but does not know that it is not permitted.

To recover from this problem we propose to replace obligations by a new operator $\mathbf{R}$ expressing agent's responsibility. The formula $\mathbf{R}_i\varphi$ means 'i is responsible for φ'.

To define agent responsibility we need some abbreviations:

$$[G{:}\delta : \mathbf{stit}]\varphi \stackrel{\text{def}}{=} [G{:}\delta]\varphi \wedge \neg\langle\!\langle \emptyset \rangle\!\rangle \varphi$$
$$[G{:}\delta : \mathbf{allows}]\varphi \stackrel{\text{def}}{=} \neg[G{:}\delta]\neg\varphi \wedge \langle\!\langle G \rangle\!\rangle \neg\varphi$$
$$[G{:}\delta : \mathbf{tries}]\varphi \stackrel{\text{def}}{=} \neg[G{:}\delta]\neg\varphi \wedge \langle\!\langle G \rangle\!\rangle \neg\varphi \wedge \neg\langle\!\langle G \rangle\!\rangle \varphi$$

Now, agent responsibility operator $\mathbf{R}$ is defined by:

$$\mathbf{R}_i\varphi \stackrel{\text{def}}{=} \mathbf{K}_i(\mathbf{O}_i\psi \wedge \langle\!\langle i \rangle\!\rangle \psi \wedge \langle\!\langle \emptyset \rangle\!\rangle \psi)$$

where ψ abbreviates $\varphi \wedge \exists\delta \mathbf{K}_i([i{:}\delta : \mathbf{stit}]\varphi \vee [i{:}\delta : \mathbf{tries}]\varphi)$.[1]

We also propose to replace the operator $\mathbf{F}$ by a new operator $\mathbf{I}$. The formula $\mathbf{I}_i(\delta)$ means 'δ is an irresponsible action for i'. The latter is defined by the following abbreviation:

$$\mathbf{I}_i(\delta) \stackrel{\text{def}}{=} \neg\mathbf{K}_i\neg([i{:}\delta : \mathbf{stit}]v_i \vee [i{:}\delta : \mathbf{allows}]v_i)$$

THEOREM 1. *The following formula is valid in CPDL. Let ψ abbreviate $\varphi \wedge \exists\delta\mathbf{K}_i([i{:}\delta : \mathbf{stit}]\varphi \vee [i{:}\delta : \mathbf{tries}]\varphi)$.*

$$\begin{array}{l} (\mathbf{K}_i(\langle\!\langle \emptyset \rangle\!\rangle (v_i \to \neg\psi) \ \wedge \mathbf{R}_i\varphi) \wedge \\ \qquad \neg\mathbf{K}_i([i{:}\delta : \mathbf{stit}]\psi \vee [i{:}\delta : \mathbf{tries}]\psi) \to \mathbf{K}_i\mathbf{I}_i(\delta). \end{array}$$

For instance, in Example 1, we can model Alice's task of keeping the balances non-negative by using $\mathbf{R}_a p$, instead of $\mathbf{O}_a p$. Then, Alice is able to decide how to behave in order to fulfil her task by checking whether $\mathbf{I}_a(\delta)$, instead of $\mathbf{F}_a(\delta)$.

4. ACKNOWLEDGMENTS

The contribution by T. de Lima and L. Royakkers is part of the research program Moral Responsibility in R&D Networks, supported by the Netherlands Organisation for Scientific Research (NWO), under grant number 360-20-160.

5. REFERENCES

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. of the ACM*, 5(49):672–713, 2002.

[2] P. d'Altan, J.-J. Meyer, and R. Wieringa. An integrated framework for ought-to-be and ought-to-do constraints. *Artif. Int. and Law*, 4:77–111, 1996.

[3] D. Harel, D. Kozen, and J. Tiuryn. Dynamic logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic Volume II – Extensions of Classical Logic*, pages 497–604. D. Reidel Publishing Company: Dordrecht, The Netherlands, 1984.

[4] A. Herzig, J. Lang, D. Longin, and T. Polacsek. A logic for planning under partial observability. In *Proceedings of AAAI 2000*, pages 768–773. AAAI Press / The MIT Press, 2000.

[5] J.-J. Meyer. A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic. *Notre Dame J. of Formal Logic*, 29(1):109–136, 1988.

[6] M. Pauly. *Logic for Social Software*. PhD thesis, ILLC, University of Amsterdam, 2001.

[7] L. M. M. Royakkers. *Extending Deontic Logics for the Formalisation of Legal Rules*. Kluwer, 1998.

[8] W. van der Hoek and M. Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75:125–157, 2003.

[1] $\exists$ abbreviates a big disjunction (the set of actions is finite).