
Artificial Intelligence and Law **7:** 69–79, 1999.

Autonomous agents with norms

FRANK DIGNUM
Fac. of Maths. & Comp. Sc., Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
E-mail: dignum@win.tue.nl

Abstract. In this paper we present some concepts and their relations that are necessary for modeling autonomous agents in an environment that is governed by some (social) norms. We divide the norms over three levels: the private level the contract level and the convention level. We show how deontic logic can be used to model the concepts and how the theory of speech acts can be used to model the generation of (some of) the norms. Finally we give some idea about an agent architecture incorporating the social norms based on a BDI framework.

Key words: agent society, norms, contracts, speech acts.

1. Introduction

In the area of Multi-Agent Systems much research is devoted to the coordination of the agents. Many papers have been written about protocols (like Contract-Net) that allow agents to negotiate and cooperate (e.g. [15]). Most of the cooperation between agents is based on the assumption that they have some joint goal or intention. Such a joint goal enforces some type of cooperative behaviour on all agents (see e.g. [3, 10]).The conventions according to which the agents coordinate their behaviour is hard-wired into the protocols that the agents use to react to the behaviour (cq. messages) of other agents.

This raises several issues. The first issue is that, although agents are said to be autonomous, they always react in a predictable way to each message. Namely their response will follow the protocol that was built-in. The question then arises how autonomous these agents actually are. It seems that they react always in standard ways to some stimulus from other agents, that can therefore determine their behaviour.

Besides autonomy, an important characteristic of agents is that they can react to a changing environment. However, if the protocols that they use to react to (at least some part of) the environment are fixed, they have no ways to respond to changes. For instance, if an agent notices that another agent is cheating it cannot switch to another protocol to protect itself. (At least this is not very common). In general it is difficult (if not impossible) for agents to react to violations of the conventions by other agents.

Related to this issue is the fact that if the conventions are hard-wired into the agent's protocols it cannot decide to violate the conventions. There might be circumstances in which the agent violates a convention in order to adhere to a private goal that it considers to be more important (more profitable). For instance, delete a file that contains a virus, while the agent should not delete files.

In this paper we will argue that deontic logic can be used to model the norms according to which agents interact with each other. Deontic logic gives the opportunity to explicitly describe the norms that can be used to implement the interactions between agents. Also it can be used to model violations of these norms and possible reactions on these violations. We refer to [7, 12, 17] for an overview of reasoning about sub-ideal states (in which an obligation is violated).

Note that deontic logic itself does not imply the reason of existence of norms, nor does it imply why agents should adhere to them. It is a modeling tool that can be used to describe the rules according to which the agents adopt norms, violate norms or adhere to them.

We distinguish three levels on which the social behaviour of an agent is determined. The highest level is that of the conventions. These conventions can be very diverse. For instance, "any request from another agent should get an answer (either positive or negative)". But also "An agent should be cooperative (if possible)". The convention level is described in Section 3.

The next level is the contract level, which is described in Section 2. Contracts describe obligations and authorizations between agents that are usually created explicitly and only hold for a limited time. An important part of this level is the description of repercussions in case of violations.

The lowest level is the private level. On this level the agent makes private judgements between different obligations and/or goals and determines the actions it will take. In this paper we will not describe the mechanism with which the agent can make this choice. We only describe the concepts on the basis of which this choice can be made. The result of the choice can be indicated as the current goal of the agent. This level is described in Section 4.

Due to space limitations we will not describe the agent architecture in which the concepts are implemented. We will suffice to say that it is based on [19]. The private part of this agent architecture can be seen as a variant of the classical BDI architecture [18]. It also resembles very closely the agent architecture of the ADEPT system described in [16]. Several intuitions about different types of norms and how they may lead to preferences are shared with Conte and Castelfranchi (e.g. [4]). The main contribution of this paper is to give a *possible* formalization of some of these notions which is used to implement an agent architecture.

2. Contracts

We will start with the contract level, because the deontic concepts that we use to model the social norms are best explained at this level. It will be shown that the

deontic concepts needed to model the other levels can be seen as special cases of the ones defined at the contract level.In our view contracts are centred around obligations and authorizations. For each obligation and authorization we indicate how it arises, how it is fulfilled (or expires) and what happens if it is violated. Not only legal contracts but also cooperation and informal agreements between agents can be described in this way. The contract describes the type of relation that exists between the agents and their mutual expectations of the behaviour of the other agent.In [20] we describe more fully how the contracts can be implemented using a formal language CoLa.

2.1. DIRECTED OBLIGATIONS

The central notion that is used to model norms on the contract level is the *directed obligation* (see e.g. [9]). It is defined as follows:

> $O_{ij}(p)$ means that agent i is obliged towards agent j (the counterparty) that p holds. $O_{ij}(\alpha)$ means that agent i is obliged towards agent j (the counterparty) that α is performed.

Note that we distinguish between obligations about situations and actions.The distinction has a practical reason. Actions that are obliged can be simply put on the "agenda", while for situations that are obliged a plan has to be devised to reach them.

Usually obligations on actions carry a time aspect indicating that the action should be performed before a certain deadline. We abstract from this feature here.

For agent i the directed obligation means that it should perform some action to fulfill the obligation. Agent j has a conditional power or *authorization* to "repair" the situation in case i does not fulfill its obligation. This means that j can in those cases demand further actions from i, cancel some of its own obligations towards i or perform some repair action himself. The directed obligations O_{ij} specify a loose coordination between two agents. It creates incentives for agent i to perform some action or reach a goal. For agent j it creates expectations about the behaviour of agent i. However, both agents are still autonomous. Agent i might decide not to perform the actions it is obliged to (if e.g. the situation changed drastically from the time the obligation arose and the costs of violating the obligation are much less than the costs of performing the action.) Only in the ideal situation all obligations will be fulfilled. In the actual world many obligations will be violated for one reason or another. In Section 4, we will sketch a first (primitive) attempt to describe some rules on the basis of which agents might decide not to fulfill an obligation. A first step towards a more general framework to describe this decision process is given in [5].

The use of deontic logic gives the opportunity to specify explicitly what should happen in these cases of violation of the obligations. We illustrate this with a small part of a contract between an airline and a passenger.

After a flight reservation an obligation exists for the airline to transport the passenger. It can be fulfilled by transporting the passenger and it is "violated" if the passenger cancels the flight or the airline cancels the flight.

In logic this is described as follows:

$$[flight - reservation(a, p)]O_{ap}(transport - passenger)$$
$$O_{ap}(transport - passenger) \rightarrow [cancel(p, ticket)]O_{pa}(pay - costs)$$
$$O_{ap}(transport - passenger) \rightarrow [cancel(a, flight)]O_{ap}(pay - costs)$$

where $[\alpha]\phi$ means that after the performance of α the formula ϕ holds. We assume here that the cancelations imply the non-performance of the obligation.

2.2. AUTHORIZATION

The concept of authorization is in some sense the counterpart of the obligation. It describes the same dependency between agents as the obligation but from the the viewpoint of the other agent. We use it in a bit wider sense nl.: If an agent has the authorization to perform some action it has some basis on which to justify it. For actions that have a deontic effect (like speech acts) authorization encompasses permission. It is not only permitted to perform the action but the deontic effect of the action is also ensured. The clearest example is that if i is authorized to demand payment from j then j is obliged to pay after the demand to do so. This is not the case if i is not authorized!

Authorizations can be generated in different ways. First they can be build in by the programmers. However, this can only be done if all agents are made by the same standards. If two agents communicate from different systems they will probably not recognize each others authorizations.

A second way to establish authorizations that is related to the first one, is the linking of authorizations to the functions or roles that agents have. E.g. a consumer agent is authorized to request prices of products. An agent that explicitly coordinates several other agents is authorized to command them to perform some task, etc.

The third way authorizations are generated is through implicit effects of actions. The effect of accepting a delivery of a product implicitly authorizes the producer to demand payment. These implicit effects of actions are defined on the convention level.

The last way to generate authorizations is by explicit creation by the agents. One agent can explicitly authorize another agent to perform some actions. We will come back to this form later on.

At the moment we model the authorization with a special predicate with two arguments: the agent and the action it is authorized to perform. This is a very simplified way to capture the authorization, which in its full form should also contain elements of time and context (in particular the source of the authorization). Another important aspect is how authorizations may be delegated to third parties.

It is not at all clear in which cases this is allowed and when it is impossible. Some work on this topic has been described in [8]. Due to space limitations we do not expand on these topics here.

We finish again with a small example of an authorization in a contract:

The airline is authorized to order the passenger to pay after a reservation has been made. The authorization finishes after the payment by the passenger. In logic this is described as:

$$[flight - reservation(p, a)]auth(a, DIRECT(a, p, pay - ticket(p)))$$
$$[pay - ticket(p)]\neg auth(a, DIRECT(a, p, pay - ticket(p)))$$

In the above formalization $DIRECT(a, p, pay - ticket(p))$ is a speech act from the airline a to the passenger p with illocution $DIRECT$, in which the airline orders (directs) the passenger to perform the action $pay - ticket(p)$.

2.3. PRACTICAL CONSIDERATIONS

Although the concept of directed obligation can be used to model all obligations on the contract level some distinctions should be made towards the implementation of these obligations. E.g. an obligation like "the customer has to pay within 3 weeks" can immediately be used by the agent to form goals and/or plans to fulfill the obligation.

Other obligations define a (vague) class of situations or actions.For instance, "agent i should cooperate with agent j". Depending on the situation this can be done in different ways. These obligations first need an interpretation to define their meaning in a concrete situation. These interpretation rules are defined on the convention level.

2.4. GENERATING OBLIGATIONS AND AUTHORIZATIONS

Obligations can be formed either through the implicit effect of an action, (defined on the convention level) or through some special type of messages.

An example of the first form is an "accept order". The acceptance of the order can imply the obligation to deliver the product. This implicit generation of an obligation stems from an existing convention. An example of an obligation arising from a speech act is a promise like: "I promise to deliver the goods on Friday" which leads to the obligation to do so. Formally:

$$[COMMIT(i, j, deliver)]O_{ij}(deliver)$$

An obligation can also arise through an authorized command. E.g. a demand for payment after the goods are delivered leads to an obligation to do so. Formally:

$$auth(i, DIRECT(i, j, pay)) \rightarrow [DIRECT(i, j, pay)]O_{ji}(pay)$$

The authorizations are created also by convention or through the special role or function of the agent or through special "authorization" messages. E.g. agent i could agree with agent j to always deliver his goods on request by giving him an authorization to ask for delivery. Formally:

$$[AUT(i, j, DIRECT(j, i, deliver))]auth(j, DIRECT(j, i, deliver)) \\ \wedge auth(j, DIRECT(j, i, deliver)) \rightarrow \\ [DIRECT(j, i, deliver)]O_{ij}(deliver)$$

For more (formal) details about these authorizations see [8].

3. Conventions

The level of conventions between agents can be compared with the *prima facie* obligations that arise from the law. Prima facie obligations hold under normal circumstances, becoming actual unless some other moral consideration intervenes ([1]). They provide a kind of "moral background" against which people (and agents) interact.The advantages of establishing some type of conventions in multi-agent environments are made clear in [11], in which it is shown that socially responsible agents perform better than selfish agents

We distinguish two types of conventions, interpretation rules and prima facie norms.

There are two types of interpretation rules. The first type are interpretation rules indicating how terms like "reasonable", "good", "cheap" are defined. E.g. Suppose that agent i should deliver computers to agent j against reasonable market prices. The convention might state that (under normal circumstances) reasonable market prices are not more than 10% above the lowest price on the market.

The second type of interpretation rules describe that certain actions will have a certain implicit (deontic) effect. E.g. If agent i orders a product from agent j then he implicitly authorizes j to demand payment upon delivery of the goods.

3.1. PRIMA FACIE NORMS

The abstract conventions are probably the best example of pure deontic sentences describing general social norms and values (also called prima facie norms). Besides the obligation we also distinguish prohibitions (indicated by F_{ij}) and permissions (indicated by P_{ij}) at this level. Usually the index indicating the counterparty is left out, because the counterparty is unknown or an abstract entity.

The prohibitions function as limitations on the behavior of agents. E.g. "Agents cannot copy information without authorization (of the owner of that information)". This is described in deontic logic as

$$\forall i \neg auth(i, copy) \rightarrow F_i(copy)$$

Crucial in this case is that there is still the possibility that the agent copies information without authorization. This will lead to a situation of violation of the norm,

but not an inconsistent state. It is also still possible to specify what should be done in this case. For instance let the agent pay a fine:

$$F_i(copy) \rightarrow [copy]O_i(pay - fine)$$

The permission operator is almost only used to indicate exceptions to a general rule or in cases of uncertainty. E.g. "persons are permitted to kill in self defense", which is an exception to the general rule that persons cannot kill other persons".

The obligations are descriptions of an ideal situation. For instance, "Agents should behave cooperatively". These types of norms cannot be transformed into goals because they are not situations that can be actually reached. However, they can be used to evaluate different possible actions and choose the most appropriate. So, in contrast to the contract level the obligations on the convention level do not (usually) give rise to actions, but are only used to choose between alternative courses of action.

In contrast to the obligations in the contracts the prima facie norms usually do not arise from actions but arise in certain situations and they remain valid as long as the situation in which they arise stays valid.

Because the prima facie norms are by definition general it can easily happen that they conflict in particular situations. For instance, "One should obey ones superior officer" and "One should not kill" will conflict when the officer commands the soldier to kill someone. Some mechanism is needed to determine which of the obligations should be followed in each actual situation. This mechanism determines the actual norms according to which an agent should behave according to the convention level. In [1] such a mechanism is described in detail and we will not go deeper into this issue here. In the next section we will see that the (remaining) actual norms of the convention level will be compared with the agents private norms and goals to determine its actual behaviour.

3.2. GENERATION AND ENFORCEMENT OF CONVENTIONS

Conventions are generally fixed when the system is started up. There are two advantages of having the convention level instead of just incorporating the conventions in the communication protocols. The first is that the conventions are now explicit and can be more easily changed (by special actions).

The second is that there can be many conflicting general conventions. These cannot be hard-wired in the agents where they would lead to contradictions. However theories like those in [1] could be incorporated into the agents in order to use the conventions.

Why would agents comply to conventions? In the most simple model this follows from the fact that the consequences of violating a convention lead to a state that is less preferred than the state that is reached by adhering to the convention. Therefore it is beneficial for the agent to adhere to the conventions.In order to enforce the adherence to conventions one could introduce some kind of police

agent which functions as the counterparty of oligations on the convention level. This agent can then take repercussions whenever agents violate the conventions. E.g. by excluding them from information.

4. The Private Level

We assume that agents have both a pro-active and a reactive behaviour. The pro-active behaviour is determined by the purpose for which the agent is designed. E.g., "collect information about agents on the WWW". And also by the build-in norms that the agent has. E.g., "I do not ly/steal/cheat".

The reactive behaviour from an agent follows from the social norms that it adheres to.

At the private level these influences come together and the agent has to merge them somehow to determine its future behaviour. We chose to translate all concepts first to a basic concept for the private level: conditional preferences (or conditional *desires* if one follows the BDI framework). The behaviour of an agent is determined completely by these preferences and their ordering.

The conditional preferences are modeled as described in [2].Conform these definitions we can define preferences as follows:

$Pref_i(\phi|\psi)$ **iff** agent i prefers ϕ to be true in every situation in which ψ is true.

These preferences are conditional to make them dependent on the environment of the agent. An agent might prefer to collect data from a large bibliography database on another continent at night and might prefer to collect its data from other sources during the day due to transport times and costs.

The build-in goals are translated into preferences automatically. (Probably only the preferences are described, without ever referring to the overall goal). All the other preferences follow from some type of obligation. This can be a build-in norm, a contract or a convention. All these norms are of the form $O_{ij}(\phi)$. If $i = j$ then it is a private (or build-in norm), if j is an abstract entity it stands for a prima facie norm and otherwise it stands for a social norm.

For a very norm abiding agent the rule that combines the obligations and preferences of the agent might look as follows:

$$\forall i, j\, Pref_i(\phi|O_{ij}(\phi))$$

I.e. in situations where an agent i has an obligation $O_{ij}(\phi)$ it prefers ϕ to be true. In general one might wish more elaborate rules to generate the preferences from obligations. Some of these are explored in [5]. Of course this does not mean that the agent will not violate this obligation. E.g. if an agent is obliged to pay for a flight that it reserved, it will prefer a situation where the flight is payed. However, it might be that the agent does not have enough money yet and does not pay anyway. So, an agent can act against a preference whenever there is a constraint preventing him to

do so or if there is another preference with higher priority (we will come back to these priorities shortly) which prevents him from doing so.

The *goals* of an agent are derived from its preferences. However, the goals of an agent are supposed to direct the behaviour of the agent. Therefore any preferences of an agent that are already true are not considered to be goals of the agent. Also preferences that cannot be "achieved" by the agent are not goals because the agent will not attempt to reach unachievable goals. In this case "achieve" is taken in a broad sense. That is, the preference cannot be achieved by the agent acting by it self but also not in cooperation with other agents.

The set of goals of an agent can thus be defined as:

> The *goals* of an agent are the preferences of an agent which are not true and which are achievable.

This is formally defined as:

$$Goal_i(\phi|\psi) \equiv Pref_i(\phi|\psi) \wedge \neg\phi \wedge Achiev_i(\phi)$$

This definition is very similar to the one given in [14] and is derived from this work and some joint work reported in [6]. Note that most preferences that stem from conventions are not goals of the agent because they are not achievable situations for the agent. However, using the above rule that generates preferences from obligations for the agent, all obligations that are not true and can be achieved will automatically turn into goals of the agent.

The above definition of a goal leaves an agent with many goals that have to be pursued at any moment in time. In order for the agent to be able to decide which goal it will pursue the goals should be ordered. This is done indirectly by ordering the states according to their (partial) fulfillment of all preferences. Like in [2] we use a utility function on the states for this purpose. The utility function contains some metric to measure how close a state is to the fulfillment of a preference and also a weight of that preference. The goal which is true in the state with the highest utility is chosen by the agent to be the first goal to be pursued. The use of a utility function for the reasoning about preferences indicates that the formal description of these inferences amounts to a non-monotonic logic.

It is important that we do not just order the goals but we consider the goals in light of the combination of all preferences. In this way we move into a direction of "general" preference instead of towards a single important goal.

It follows from the above that agents that behave according to this utilitarian principle will decide to fulfill all obligations that give them the highest utility. That is, an agent will violate an obligation only if it has another goal with a higher utility.

Of course, the way an agent behaves is determined for a large part by the definition of the utility function. If some conventions indicating cooperative behaviour get a high weight then the agent will tend to behave cooperatively. Also when preferences stemming from contracts of the agent get high weights the agent will react

quickly on social obligations. This leads to open minded agents in the terminology of Kinney and Georgeff [13].

5. Conclusions

In this paper we have given an overview of the concepts that are used to model the social norms that govern the behaviour of autonomous agents. We have shown that many interrelated concepts are needed to capture (part of) the behaviour of autonomous agents. Most of these concepts can be modeled using deontic notions. The use of deontic logic captures both the autonomy of the agents as well as the (social) dependencies between agents. Dividing the concepts over three levels makes it possible to structure the different social interactions of an agent. Although the (directed) obligation plays a central role it is accompanied by different concepts on each level and takes a slightly different form on each level as well. We have given some indications towards the implementation of the norms into a multi-agent system where the agents are based on a type of BDI architecture. Due to shortage of space many questions with regard to an actual implementation remain open.

References

1. N. Asher & D. Bonevac 1996. Prima facie obligation, *Studia Logica* **57**(1), 19–45.
2. C. Boutilier 1994. Toward a logic for qualitative decision theory, in Jon Doyle et al. (eds.), *Proceedings of the 4th Int. Conf. on Principles of Knowledge Representation and Reasoning*, San Francisco (CA): Morgan Kaufmann, pp. 75–86.
3. P. Cohen & H. Levesque 1991. Teamwork *Nous* **35**, 487–512.
4. Conte, R. & C. Castelfranchi 1995. *Cognitive and Social Action*. London: UCL Press.
5. F. Dignum & R. Conte 1997. Intentional agents and goal formation, in M. Singh et al. (eds.), *ATAL-97*, Providence, USA, pp. 219–231.
6. F. Dignum & B. van Linder 1997. Modelling social agents: Communication as actions, in M. Wooldridge J. Muller & N. Jennings (eds.), *Intelligent Agents III (LNAI-1193)*, Springer-Verlag, pp. 205–218.
7. F. Dignum, J.-J.Ch. Meyer, & R. Wieringa 1994. A dynamic logic for reasoning about sub-ideal states, in J. Breuker (ed.), *ECAI Workshop on Artificial Normative Reasoning*, Amsterdam, pp. 79–92.
8. F. Dignum & H. Weigand 1995. Modeling communication between cooperative systems, in J. Iivari, K. Lyytinen & M. Rossi (eds.), *Advanced Information Systems Engineering*, Berlin: Springer, pp. 140–153.
9. H. Herrestad & C. Krogh 1995. Deontic logic relativised to bearers and counterparties, in J. Bing & O. Torrund (eds.), *Anniversary Anthology in Computers and Law*, Tano A.S., pp. 453–522.
10. N. Jennings 1993. Commitments and conventions: The foundation of coordination in multi-agent systems, *Knowledge Engineering Review* **8**(3), 223–250.
11. N. Jennings & J. Campos 1997. Towards a social level characterisation of socially responsible agents, *IEEE Proceedings on Software Engineering* **144**(1), 11–25.
12. A. Jones & I. Pörn 1985. Ideality, sub-ideality and deontic logic, *Synthese* **65**, 275–290.
13. D. Kinny & M. Georgeff 1991. Commitment and effectiveness of situated agents, In *Proceedings IJCA Intelligence*, Sydney: Australia, pp. 82–88.

14. B. van Linder, W. van der Hoek & J.-J.Ch. Meyer 1996. How to motivate your agents. On making promises that you can keep, in Wooldridge, Müller & Tambe (eds.), *Intelligent Agents II*, LNCS 1037, pp. 17–32.
15. J. Muller 1996. A cooperation model for autonomous agents, In J. P. Müller, M. J. Wooldridge, & N. R. Jennings (eds.), *Intelligent Agents III – Proceedings of (ATAL-96)*, Lecture Notes in Artificial Intelligence, Heidelberg: Springer-Verlag.
16. T. J. Norman, N. R. Jennings, P. Faratin, & E. H. Mamdani 1996. Designing and implementing a multi-agent architecture for business process management, in J. P. Müller, M. J. Wooldridge, & N. R. Jennings (eds.), *Intelligent Agents III – Proceedings of (ATAL-96)*, Lecture Notes in Artificial Intelligence. Heidelberg: Springer-Verlag.
17. H. Prakken & M. Sergot 1996. Contrary-to-duty obligations, *Studia Logica* **57**(1), 91–115.
18. A.S. Rao & M.P. Georgeff 1991. Modeling rational agents within a BDI-architecture, in J. Allen, R. Fikes & E. Sandewall (eds.), *Proceedings 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning*, San Mateo (CA): Morgan Kaufmann, 473–484.
19. E. Verharen, F. Dignum, & H. Weigand 1996. A language/action perspective on cooperative information agents, in E. Verharen, N. van der Rijst & J. Dietz (eds.), *Proceedings International Workshop on Communication Modeling (LAP-96)*, Oisterwijk: The Netherlands, pp. 40–53.
20. H. Weigand, E. Verharen, & F. Dignum 1996. Interoperable transactions in business models: A structured approach, in P. Constantopoulos, J. Mylopoulos & Y. Vassiliou (eds.), *Advanced Information Systems Engineering (LNCS 1080)*, Springer, pp. 193–209.