# Deliberative Normative Agents: Principles and Architecture

Cristiano Castelfranchi[1], Frank Dignum[2], Catholijn M. Jonker[3], Jan Treur[3]

[1] National Research Council - Institute of Psychology
Division of AI, Cognitive Modelling and Interaction,
PSCS - Social Simulation Project, Viale Marx, 15-00137 Roma, Italy
Email: cris@pscs2.irmkant.rm.cnr.it

[2] Eindhoven University of Technology,
Faculty of Mathematics and Computer Science
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
Email: dignum@win.tue.nl URL : http://wwwis.win.tue.nl/~dignum

[3] Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Email: {jonker, treur}@cs.vu.nl
URL: http://www.cs.vu.nl/{~jonker,~treur}

**Abstract**

In this paper norms are assumed to be useful in agent societies. It is claimed that not only following norms, but also the possibility of 'intelligent' norm violation can be useful. Principles for agents that are able to behave deliberatively on the basis of explicitly represented norms are identified and an architecture is introduced. Using this agent architecture, norms can be communicated, adopted and used as meta-goals on the agent's own processes. As such they have impact on deliberation about goal generation, goal selection, plan generation and plan selection.

## 1 Introduction

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 impredictable 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). What we believe to be necessary is an autonomous normative agent, able to take into account the existence of social norms in its decisions (either to follow or violate a norm) and able to react to violations of the norms by other agents.

Obviously, if the conventions and norms are hard-wired into the agent's protocols it cannot decide to violate the norms. On the contrary, 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 has the norm that it should not delete files.

In order to address the above issues we propose to have *deliberative normative* agents. Deliberative normative agents are agents that have explicit knowledge about the enacted norms in a multi-agent environment and can make a choice whether to obey the norms or not in specific cases. Of course this requirement also has consequences for the architecture of the agent. How do norms influence the behaviour of the agent? As a minimal prerequisite we consider that the agent should be a cognitive agent. That is, it should have some representation of some mental attitudes like beliefs, goals and intentions; e.g., the BDI-architecture [20].

The norms should in some way influence the behaviour of the agent. However, they cannot be incorporated as some filter on the possible goals or constraints on the decision process. In that case the agent would always obey the norms (if possible), while we want the decision to obey the norm to be a motivated 'conscious' separate decision. So, the architecture should allow for some facility for reasoning about applying the norms and subsequent combination of the result with the goals and plans of the agent. The combination of goals, plans and norms thus determines the actual behaviour of the agent.

In this paper we describe a generic architecture for deliberative normative agents. The architecture includes specific components that manage the norms and the interaction between norms, goals and plans. In Section 2 the architecture is described globally and a more detailed justification is given of the components. In Section 3 the relations between norms and actual behaviour are discussed: the relations between norms and goals are described in more detail, and the relations between norms and plans.

## 2 Global Description of the Architecture

In Section 2.1 the assumptions behind the architecture for deliberative normative agents are discussed. Next, in Section 2.2 the architecture is explained at two levels of process abstraction, and examples are given as an illustration.

### 2.1 Principles Behind the Architecture

The architecture we aim at depends on the kind of social and normative agent (and behaviour) we want. Our objective is a norm-autonomous agent; i.e., an agent

- able *to know that a norm exists in the society* and that it is not simply a diffuse habit, or a personal request, command or expectation of one or more agents;
- able *to adopt this norm* impinging on its own decisions and behaviour, and then

- able *to deliberatively follow* that norm in the agent's behaviour, but also
- able *to deliberatively violate* a norm in case of conflicts with other norms or, for example, with more important personal goals; of course, such an agent can also accidentally violate a norm (either because it ignores or does not recognize it, or because its behaviour does not correspond to its intentions).

To adopt a norm does not necessarily imply to follow it. The concept of 'adopting a norm' means that the agent decides to generate goals and plans on the basis of its belief that there is such a norm andthat it also concerns the agent itself. These are the 'normative beliefs' and the generated goals are the 'normative goals'. However, although I have adopted the norm and generated the corresponding goals, these goals will not necessarily always be preferred to my other active goals: they will not necessarily become one of my intentions and determine my external behaviour. Thus, although I adopt a norm I can deliberatively violate it.

Norms cannot be simple static constraints on behaviour or on decisions: the agent's goals and preferences, its decisions among conflicting goals, and the agent's plans must be based on its beliefs (reasons) and norms. In other words a deliberative normative agent is a cognitive agent that bases its behaviour on, for example, goals, beliefs, intentions, plans, or decisions. This kind of normative agent is a norm-follower, it can conform its behaviour to social or legal norms, but it can also be a cheater, an opportunistic agent that is violating a norm case by case when this is convenient, or it can be a rebel violating norms for principled (moral or political) disagreements or for being against norms in general. This kind of agent cannot be simply controlled from the outside or rigidly commanded. However it can be *influenced* and induced to do or refrain from doing something if, e.g., an authority issues a norm that concerns it, or another agent informs it about the existence of such a norm. In other words, what we aim at is not only the possibility to follow norms, but also 'an *intelligent* violation of norms'.

Agents should also be able to collectively issue norms, to reason, communicate and negotiate about them. Thus norms cannot simply be implicitly represented constraints in the agents architecture or an external fixed rule; they must be also *mental objects*; there must be some mental representation of them [9], [11], [12], [14]. In fact in this architecture norms are mental representations entering the mental processing, interacting in several ways with beliefs, goals, and plans and thus are able to determine the agent's behaviour. In Sections 2.2 to 2.4 the deliberative normative agent architecture is described globally.

### 2.2 The Top Level Within the Agent

The architecture for deliberative normative agents introduced here has been designed as a refinement of the generic agent model presented in [7] . Compared to this generic agent model, in Fig. 1 the following differences can be found. The components that the current architecture has in common with the generic agent model are Agent Interaction Management, World Interaction Management, Maintenance of Agent Information, Maintenance of World Information, and Own Process Control.

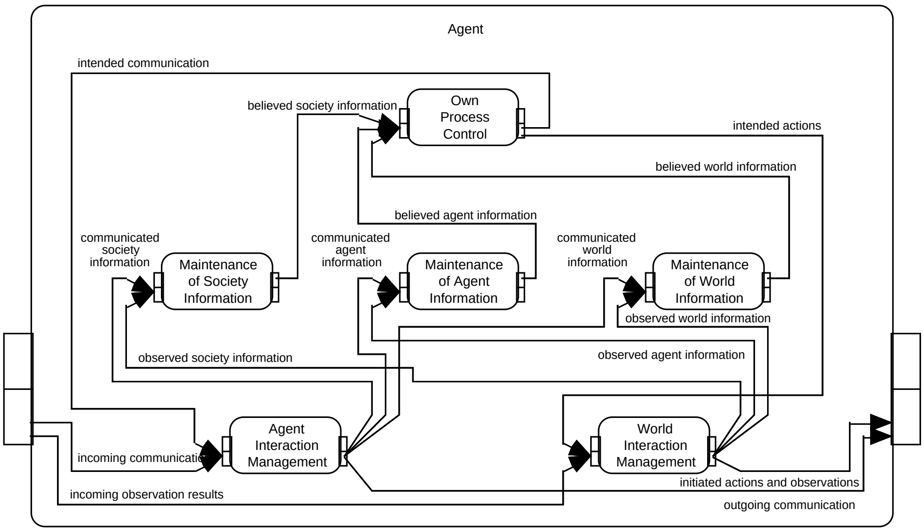


**Fig. 1.** Top level within the agent

The generic agent model further includes the components Cooperation Management, Maintenance of History, and Agent Specific Tasks. These components have been omitted in the current architecture for briefness; they can be added when needed. Furthermore, the current model includes the component Maintenance of Society Information; this component is not part of the generic agent model as presented in [7]. This component is added specifically for dealing with society properties such as norms.

Information about aspects external to the agent can be received by communication or by observation (perception). If the agent decides that information is valuable, it is stored. In storing information within the agent a distinction is made according to the content of the information, abstracting from the source of information. Information about the world (the agent's *world model*) is stored in the component Maintenance of World Information. Information about other agents (the agent's *agent models*, sometimes also called *acquaintance models*) is stored in the component Maintenance of Agent Information. Finally, information about the society as a whole (the agent's *society model*) is stored in the component Maintenance of Society Information.

### 2.3 Representation at Different Meta-levels Within the Architecture

To reflect semantical distinctions between different types of information within the agent, an object level and two meta-levels have been introduced.

#### Object level

In the generic model the information believed by the agent is represented as *object level* information. For example, the belief of the agent that society1 is heterogeneous (i.e., consists of different types of agents) can be represented in the component Maintenance of Society Information as

has_society_type(society1, heterogeneous)

Society norms are also explicitly represented as a specific type of society information; for example, ‘you ought to drive on the right’ or ‘be altruistic’ as a society norm can be stored in the component Maintenance of Society Information, represented as

has_norm(society1, be_altruistic)
has_norm(society1, you_ought_to_drive_on_the_right)

The three maintenance components (Maintenance of World Information, Maintenance of Agent Information, and Maintenance of Society Information) are all at the object level.

**Meta-level**

In processing incoming and outgoing information (by communication or observation), the process events involved are represented at a meta-level, with reference to the information involved. For example, if another agent B has communicated that it has you_ought_to_drive_on _the_right as a norm, then at the input interface of the agent this is represented by:

communicated_by(has_norm(agent_B, you_ought_to_drive_on_the_right), positive_assertion, agent_B)

Here positive_assertion denotes the illocution. Note that the choice to explicitly represent norms as mental concepts (in contrast to, e.g., norms as constraints outside the mental processing) makes it possible to have communication about norms. Via the information link incoming communication (see Fig. 1), this communication information is transferred to the component Agent Interaction Management, in which the content information is extracted from the communication information. In this example, within this component it is identified that the content information is agent information (i.e., information about agent B), and not, for example, society information. Based on knowledge of the form (where A:AGENT and N:NORM are variables over sorts AGENT and NORM, respectively)

**if** belief(reliable(A:AGENT), pos)
**and** communicated_by(has_norm(self, N:NORM), positive_assertion, A:AGENT)
**then** new_agent_info(has_norm(A:AGENT, N:NORM), pos)

the conclusion

new_agent_info(has_norm(agent_B, you_ought_to_drive_on_the_right), pos)

is derived. If however, agent B communicates that you_ought_to_drive_on_the_right is a norm in society1, and the agent B is considered reliable, then it is derived

new_society_info(has_norm(society1, you_ought_to_drive_on_the_right), pos)

For example, if agent B is a police agent, it may be considered reliable automatically. By knowledge of the form

```
if      belief(police_agent(A:AGENT), pos)
and     communicated_by(has_norm(society1, N:NORM), positive_assertion, A:AGENT)
then    new_society_info(has_norm(society1, N:NORM), pos)
```

the above conclusion can be derived within the component Agent Interaction Management. All these statements on the communication process are represented at the meta-level with respect to content statements such as

```
has_norm(society1, you_ought_to_drive_on_the_right)
```

By the information link communicated society information (see Fig. 1), the meta-information

```
new_society_info(has_norm(society1, you_ought_to_drive_on_the_right), pos)
```

is transferred to the component Maintenance of Society Information, and within this component it is reflected downwards and stored at the object level. These facts are considered and used as the agent's beliefs, and if norms are concerned, normative beliefs. If such an object level fact is used as a belief in the agent's own internal processes, it is represented as the meta-level information:

```
belief(has_norm(society1, you_ought_to_drive_on_the_right), pos)
```

Note that in the knowledge specification above some specific belief statements already occurred: the belief that an agent is reliable, or the belief that an agent is a police agent. The components World Interaction Management, Agent Interaction Management and Own Process Control are at the meta-level; within Own Process Control also meta-meta-level reasoning is involved.

**Meta-meta-level**

Since norms have an effect on the control of the agent's internal processes, within Own Process Control also a meta-meta-level is used to explicitly represent information used to reason about control on the agent's own internal processes. This will be discussed in more detail in Section 3.

### 2.4 Global Structure of Own Process Control

The component Own Process Control within the architecture is further refined: as depicted in Fig. 2, Own Process Control is a composed component in the current architecture. The component Norm Management determines which norms the agent is adopting for itself and in what ways the agent wants its behaviour to be influenced by norms (adopted, and rejected). On the basis of norms meta-goals are created that influence the strategies used by the agent. The component Strategy Management, therefore, uses norms to determine the strategies with which goals and plans are formed. On the basis of these strategies the component Goal Management determines

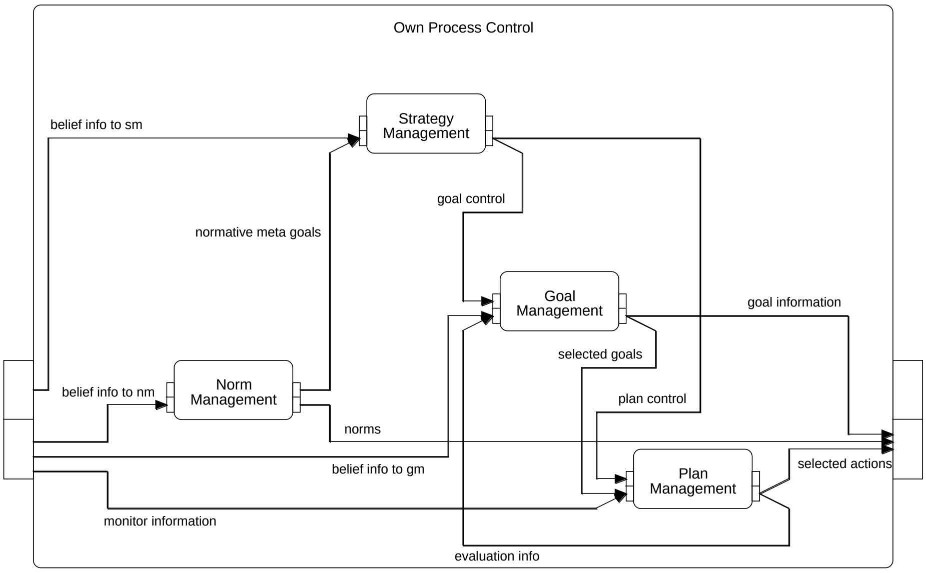


**Fig. 2**. Top level within Own Process Control

which goals the agent wants to pursue, and the component Plan Management determines plans for the current goals of the agent. Normative beliefs both for adopted and non-adopted norms may play a role in the determination of these goals and plans as well. In Section 3 the functioning of Own Process Contol will be discussed in more detail.

In Section 2.3 it was shown in an example how the society norm you_ought_to_drive_on_the_right as communicated by a police agent leads to an agent's belief that the norm you_ought_to_drive_on_the_right is a society norm in society1, stored in the component Maintenance of Society Information. Via the information links believed society information (see Fig. 1) and belief info to nm (see Fig. 2), the component Norm Management, which is at the meta-level within Own Process Control, receives as input

belief(has_norm(society1, you_ought_to_drive_on_the_right), pos)

In case the agent considers itself as belonging to society1, it can decide to try to conform to such a society norm, using the knowledge

| | |
|---|---|
| **if** | belief(has_norm(society1, N:NORMS), pos) |
| **and** | belief(belongs_to(self, society1), pos) |
| **then** | adopted_norm(N:NORM) |

For example, using the beliefs belief(has_norm(society1, you_ought_to_drive_on_the_right), pos) and belief(belongs_to(self, society1), pos) it can derive

adopted_norm(you_ought_to_drive_on_the_right)

This output of Norm Management is transferred to input of the component Strategy Management. This component, which is at the meta-meta-level, uses the representation

adopted_own_process_goal(you_ought_to_drive_on_the_right)

The information link normative meta-goals specifies that semantically speaking an adopted norm corresponds to a goal on the agent's own internal processes, i.e., a meta-goal or own process goal. How these own process goals affect the agent’s internal functioning is addressed in more detail in Section 3.

The component Goal Management is composed of two sub-components: Goal Generation (where candidates for goals are generated) and Goal Selection (where a choice is made from the candidate goals). In a similar manner Plan Management is composed of Plan Generation and Plan Selection. Action generation and selection is part of plan generation and selection.

## 3 Norms and Behaviour

As discussed, norms are represented by mental objects entering the mental processing, that interact with beliefs, goals and plans. In other words norms are crucial in the functioning of a normative cognitive agent. Let us see how norms - in particular when adopted - impact (thanks to their mental implementation) on the mental process governing the behaviour. Eventually in fact, the aim of norms is to determine the behaviour of agents in the society, groups and organisations; thus we have to show how a norm impinging on a given agent eventually can determine its behaviour and can produce or avoid a specific act. In the architecture introduced here the choice has been made that the agent generates behaviour by generating and selecting goals on the basis of beliefs and norms, and generating and selecting actions and plans on the basis of the selected goals. In the next two sections the impact of norms on goal determination and on actions and plan determination is discussed in more detail.

### 3.1 Impact of Norms on Goals

In our architecture *adopted norms* play two main roles in goal determination, according to the composition of the component Goal Management introduced in Section 2.4 (in terms of the components Goal Generation and Goal Selection).

- Norms have impact on *goal generation*; goals that do not derive from desires or wishes: what we should/have to do, not what we would like/wish to do. Thus norms are among the possible ‘*sources of goals*’ together with bodily needs, emotions, wishes and desires, and others.

By generating goals, norms provide also a target for reasoning. In a goal directed agent in fact the reasoning activity is no longer spreading without direction; it can be goal-driven, focused on the problem. Norms can provide this orientation to reasoning, by specifying which normative goal we should take into account.

- Norms have impact on *goal selection* by providing criteria about how to manage and select among existing goals; in particular preference criteria. They push us to prefer one goal over another. For example, the norm 'be altruistic!' or the norm 'obey to norms!' do not specify and provide the agent any specific behaviour or goal (like 'you ought to stop for a red traffic light!'), they provide a criterion about what goal should be preferred.

In the agent model, the agent's own processes are specified in terms of a functional architecture, describing what output is generated over time, given input over time. The impact of own process goals in the form of control of the functionality of the agent's own processes can be effectuated in three manners:

1. an own process goal has impact on the *input* of the functioning of the agent's own processes; e.g., it defines a focus on the input information used in the processing
2. an own process goal has impact on what *output* is attempted to generate; e.g., it defines a focus on the targets to be directed to in the processing
3. an own process goal defines a focus on the *functionality relation* between input and output; e.g., it defines a focus on the part of the knowledge to be used in the specification of the functionality relation

All three types of focussing are possible within the model. To determine this focussing is the task of the component Strategy Management. In this section examples are discussed for the goal management process. In Section 3.2 examples are discussed for the plan management process.

### Examples of strategy determination

A strategy can be defined by a number of elements of the three types listed above. An example of the first type is the generation of presuppositions:

| | |
|---|---|
| **if** | adopted_own_process_goal(maximize_own_property) |
| **and** | belief(is_available(P:PRODUCT), pos) |
| **then** | presupposition(goal_source(owns(self, P:PRODUCT), pos), maximize_own_property) |

The effect of this is that the presuppositions goal_source(owns(self, P:PRODUCT) are actually used in the processing of Goal Generation and lead to candidate goals to achieve owns(self, P:PRODUCT) for all possible products P.

An example of the second type is when from an adopted own process goal it is derived which specific normative goals should be taken into account in the goal generation process, specified as follows:

| | |
|---|---|
| **if** | adopted_own_process_goal(N:NORM) |
| **and** | is_in_context_of(G:INFO_ELEMENT, N:NORM) |

**then** to_be_determined(candidate_normative_goal_for(G:INFO_ELEMENT, S:SIGN, N:NORM))

An example of the third type is when it is derived which selection criteria should be used in the goal selection process:

**if** adopted_own_process_goal(N:NORM)
**and** is_criterion_for(C:CRITERION, N:NORM)
**then** functionality_element(selection_criterion_for(C:CRITERION))

**The use of strategy within Goal Management**

The conclusions derived within Strategy Management of the form

presupposition(admissible_goal_source(owns(P:PRODUCT), pos), N:NORM)
to_be_determined(candidate_normative_goal(G:INFO_ELEMENT, S:SIGN, N:NORM))
functionality_element(selection_criterion_for(C:CRITERION))

are transferred by the information link goal control to the component Goal Management. Within Goal Management the first two types are transferred to Goal Generation, the third to Goal Selection. Within Goal Generation a presupposition

presupposition(admissible_goal_source(owns(P:PRODUCT), pos), N:NORM)

defines a set of input goal sources, and the information

to_be_determined(candidate_normative_goal(G:INFO_ELEMENT, S:SIGN, N:NORM))

defines the targets for the reasoning process of the form

candidate_normative_goal(G:INFO_ELEMENT, S:SIGN, N:NORM)

The precise knowledge by which goals are generated depends on the application addressed. The generic deliberative normative agent model only provides elements that can be used; it does not commit to a specific approach to Goal Generation.

Within Goal Selection, goals (normative and other) are compared. In case of a goal conflict the resolution of this conflict uses the normative selection criteria (if any) transferred from Strategy Management. Within Goal Selection, the normative goals may get priority over the non-normative goals in conflict with them. However, it is also possible that the normative goals do not get priority; in this case the agent deliberatively violates the norms. Also the precise knowledge within Goal Selection will depend on the application addressed.

### 3.2 Impact of Norms on Actions and Plans

In general, goals have a crucial impact on the process of plan generation and selection. In the case of a deliberative normative agent, where norms have an impact on the goals that are generated and selected, in an indirect manner norms have impact on plans as well. In addition to this impact, also a direct impact is possible, especially in cases where norms indicate more that certain actions are not done, than that they indicate certain goals. For example, the norm ‘always use the most friendly, least aggressive means to achieve your goals’ refers to properties of actions and plans instead of specific types of goals. Similar to the previous section, in our architecture *adopted norms* have two types of direct impact in Plan Management:

- Norms may have impact on *plan generation*: an adopted norm may lead to a focus on generation of specific types of actions and plans, and exclude certain other actions and plans to be generated at all.
- Norms may have impact on *plan selection* by providing criteria about how to manage and select among existing plans, in particular preference criteria. They push us to prefer one plan (for the selected goals) to another. For example, the norm ‘be kind to colleagues!’ provides a criterion about what action should be preferred among different possible plans to achieve a goal within an organisation.

The three forms of impact introduced in the previous section also apply here. An example of the first kind is the generation of presuppositions:

```
if      adopted_own_process_goal(minimize_damage)
and     belief(is_clean_plan(P:PLAN), pos)
then    presupposition(admissible_plan(P:PLAN), minimize_damage)
```

The effect of this is that the presuppositions admissible_plan(P:PLAN) are actually used in the processing of Plan Generation and lead to candidate plans.

An example of the second type is when from an adopted own process goal it is derived which specific plans should be taken into account in the plan generation process, specified as follows:

```
if      adopted_own_process_goal(N:NORM)
and     is_in_context_of(P:PLAN, N:NORM)
then    to_be_determined(candidate_normative_plan(P:PLAN, N:NORM))
```

An example of the third type is when it is derived which selection criteria should be used in the plan selection process:

```
if      adopted_own_process_goal(N:NORM)
and     is_criterion_for(C:CRITERION, N:NORM)
then    functionality_element(plan_selection_criterion_for(C:CRITERION))
```

As in the case of Goal Generation and Goal Selection, the impact of these strategy elements on Plan Generation and Plan Selection can be specified depending on the application addressed. Also here it is possible to deliberatively resolve conflicts within Plan Selection either following norms (giving candidate normative plans highest priority) or violating norms (giving candidate normative plans not highest priority).

## 4 Discussion

There is an extensive literature about agent theories concerning beliefs, goals and intentions. However, there is not much theory available to incorporate norms into the behaviour of agents [13]. On the one hand, there is work on normative agents but of an experimental nature and for the purpose of social simulation. In this type of work agent societies are compared where in one society the agents behave selfish, while in another they behave altruistic. In these types of experiments the norms are built into the agent. The agent cannot change its behaviour over time, based on experience. On the other hand, there are more complex normative agents for multi-agent systems, mainly with the purpose of reducing coordination or transaction costs but in these agents norms are simply built-in constraints in the agent's architecture [22], [23] or rules and protocols the agent necessarily applies [17].

Boman [2] interestingly introduces norms in his agent architecture to overcome serious limitations of rational decision making, for example in order to have a threshold of unacceptable damages, or to take care of the advantages of the group (this is close to what Jennings and Campos in [18] attempt to do). However, in this architecture norms act only from outside the decision maker: they do not generate goals or meta-criteria to be taken into account during the decision making. Either they simply modify the decision parameters, or they post hoc filter decisions and actions. Thus, we can neither say that norms are explicitily represented and reasoning about them takes place, nor that the agent deliberates to follow or violate a norm. The agent cannot really violate a norm, which is in fact just a complex constraint. As for [18] they take into account, for example, the collective interest in the agent's decision, but they do not account for the normative origin and character of this goal: it is simply a pro-social attitude of the agent.

An alternative to the approach introduced are utility-driven agents. They make decisions among different behavioural alternatives on the basis of utility and probability [2], [18], [21]. All of the approaches described in [24], [16], [1], [3] are utility based. All of these approaches describe some social attitudes of agents. The approach described in [24] uses "motivational quantities" to describe them, while the other approaches use an explicit social component in the agent's utility function. Although we recognize these approaches as useful practical solutions for describing social influences, the utility based approaches all suffer from some drawbacks. First of all the weight ascribed to the social utility for each agent is fixed. This means that an agent will find the social aspect of its utility equally important during its whole life. It cannot change this weight depending on its experiences! A second drawback is that it is very difficult to describe the effect of a violation of an obligation in this framework. It is obvious that such a violation has an immediate impact on the utility that other agents ascribe to e.g. cooperating with this agent, but it is unclear how this relation can be made within the framework. Ofcourse the utility based frameworks have the advantage that preferences between attitudes can be modelled (see [3]) and a comparison can be made between them (through the weighting factors). However, none of the papers gives any theory on how to assign these preferences or weights for the different components of the utility function. The work described in [26] is interesting in that it describes the complementary viewpoint from our work. It looks at the expectancy an agent has that another agent will keep its commitment and the influence of this expectance on its decision to perform an action. In our work we look at the influence of the obligation on the decision of the agent itself.

Conte and Castelfranchi proposed in [9], [11] a *cognitive approach* to norms in artificial agents, where norms are conceived as external (expectations, behaviours, and prescriptions) and internal (i.e., mental) entities. They show how norms are acknowledged and issued by the agents, and how they are translated into as normative believes and produce normative goals. They also characterise different kinds of norm-adoption (parallel to goal-adoption) based on different attitudes and motives about adopting the norm. However, on the one hand, the formalisation of this process based on the approach of Cohen and Levesque [8] is partial, and on the other hand they do not insert this model of norm-processing in some operational architecture. For example, in their simulation experiments about norm functionalities although the theory of norms was based on explicit mental representations, they used simple reactive agents with a given normative behaviour.

Of course, an important theory that could be used to incorporate norms into the agent theory is that of deontic logic [15], [19], [25]. A first attempt has been made in [14]. In this work several types of norms are distinguished and translated into *obligations* for the agent. All the obligations result into conditional goals for the agent. The decision whether to comply to a norm or not is made by ranking the goals. If the goal resulting from a norm is ranked on top the norm will be complied with, otherwise it might be violated. The theory does not provide an explicit reasoning about complying to a norm or violating it, nor does it provide an operational architecture.

In this paper an explicit model is introduced for norm-processing within an autonomous deliberative agent; their relations with beliefs, decisions, goals, plans, and actions. In other terms, a process model is presented formalizing how norms succeed in influencing the agent's behaviour, although being possibly violated. However, we have examined only one side of the problem, i.e. the *generative* function of norms: norms activate specific goals that can become intentions (*goal and plan generation*), and then are externally executed, norms favour one goal/behaviour/intention to another (*goal and plan selection*). Thus overt behaviour can 'follow the norm'. However this is everything but a trivial notion (as is well known in philosophy) and we didn't in fact consider this side of the relation between norms and behaviour: the evaluative function of norms.

A normative agent must also be able *to check* whether a given behaviour (either its own or that of other agents) is or is not conform the norm. In particular, *a respectful agent also wants the others to be respectful* [9]. Moreover, to know whether the others (and how many of them) follow the norm can be a criterion or an *incentive for following or violating* it. Finally, there are several different normative roles and for sure at least the police agent must be able to match the behaviour of the other agents against the norm. When the norm specifies and prescribes (permits or prohibits) a given action, this check is not so difficult; but, when the norm just provides a criteria of choice (e.g., 'be altruistic') the problem is much more complex.

We have to figure out the decision process of another agent and evaluate whether in its decision making it did or did not apply these criteria.

One of the objectives of modelling explicit normative reasoning and decisions in a compositional manner in DESIRE [6] (for a real-world case study, see [5]) is that of using this distributed platform for doing social simulation about the functions of norms in multi-agent systems. For a society of simple non-cognitive agents such experiments using DESIRE were reported in [4]. The aim is to have social simulation with complex cognitive agents, able to intelligently violate norms or to change their behaviour depending on their evaluation of the situation or of the partners. We (together with Rosaria Conte) in particular plan to conduct experiments about different kinds (more or less decentralised) of normative social control and issuing, different kinds of normative 'personalities' in agents, and about their effects.

## Acknowledgements

We whish to acknowledge Rosaria Conte's contribution to a preliminary definition of this normative architecture. The text was improved on the basis of some comments of Leon van der Torre and Magnus Boman who have read an earlier version of the paper.

## References

1. Boella, G., Damiano, R., and Lesmo, L. (2000). Cooperating to the group's utility. In: N.R. Jennings, Y. Lesperance (eds.), Intelligent Agents VI. Proc. of the Sixth International Workshop on Agent Theories, Architectures and Languages, ATAL'99. Lecture Notes in AI, Springer Verlag. This volume.
2. Boman, M. (1997). Norms as Constraints on Real_time Autonomous Agent Action. In: M. Boman, W. van de Velde (eds.), Multi-Agent Rationality, Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi- Agent World, MAAMAW'97, Lecture Notes in AI, vol. 1237, Springer Verlag, Berlin, 1997, pp. 36-44.
3. Brainov, S. (2000). The role and the impact of preferences on multi-agent interaction. In: N.R. Jennings, Y. Lesperance (eds.), Intelligent Agents VI. Proc. of the Sixth International Workshop on Agent Theories, Architectures and Languages, ATAL'99. Lecture Notes in AI, Springer Verlag. This volume.
4. Brazier, F.M.T., Dunin Keplicz, B., Jennings, N., and Treur, J. (1997). DESIRE: Modelling Multi-Agent Systems in a Compositional Formal Framework. International Journal of Cooperative Information Systems, vol. 6, Special Issue on Formal Methods in Cooperative Information Systems: Multi-Agent Systems, (M. Huhns and M. Singh, eds.), 1997, pp. 67-94. Preliminary shorter version in: Lesser V. (ed.), Proceedings of the First International Conference on Multi-Agent Systems, ICMAS'95, MIT Press, Menlo Park, VS, 1995, pp. 25-32
5. Brazier, F.M.T., Eck, P.A.T. van, and Treur, J. (1997). Modelling a Society of Simple Agents: From Conceptual Specification to Experimentation. In: Conte, R., Hegselmann, R. and Terna, P. (eds.) Simulating Social Phenomena, Proc. of the International Conference on Computer Simulations and Social Sciences, ICCS&SS'97, Lecture Notes

in Economics and Mathematical Systems, Vol. 456, Springer-Verlag, Berlin, 1997, pp. 103-107.

6. Brazier, F.M.T., Jonker, C.M., and Treur, J. (1998). Principles of Compositional Multi-agent System Development. In: J. Cuena (ed.), Proceedings of the 15th IFIP World Computer Congress, WCC'98, Conference on Information Technology and Knowledge Systems, IT&KNOWS'98, 1998, pp. 347-360.
7. Brazier, F.M.T., Jonker, C.M., and Treur, J. (1999). Compositional Design and Reuse of a Generic Agent Model. In; B. Gaines, M. Musen (eds.). Proceedings of the Knowledge Acquisition Workshop, KAW'99, Banff, 1999. URL: http://sern.ucalgary.ca/KSI/KAW/KAW99/papers/Brazier1/KAW_brazier.pdf.
8. Cohen, P., and Levesque, H. (1991). Teamwork, Nous, vol.35, pp. 487-512.
9. Conte, R., and Castelfranchi, C., (1995a). Cognitive and Social Action. UCL Press, London.
10. Conte, R., and Castelfranchi, C. (1995b). Understanding the effects of norms in social groups through simulation. In Artificial societies: the computer simulation of social life. Eds. G.N. Gilbert and R. Conte, London, UCL Press.
11. Conte, R. and Castelfranchi, C. (1996). From Conventions to Prescriptions: Toward an integrated Theory of Norms, ModelAge'96 Workshop, Sesimbra, January 1996 (AI&Law, forthcoming).
12. Conte, R., Castelfranchi, C., and Dignum, F. (1999). Autonomous Norm-Acceptance, In: J.P. Mueller, M.P. Singh, A.S. Rao (eds.), Intelligent Agents V, Proc. of the Fifth International Workshop on Agent Theories, Architectures and Languages, ATAL'98. Lecture Notes in AI, vol. 1555, Springer Verlag, Berlin, Springer Verlag.
13. Conte, R., Falcone, R., Sartor, G. (1999). Agents and norms: How to fill the gap? In: Conte, R., Falcone, R. and Sartor, G. (eds) Agents and Norms, special issue, AI&Law, Vol.7, No.1, pp. 1-15, 1999.
14. Dignum, F. (1996). Autonomous Agents and Social Norms. In: R. Falcone and R. Conte (eds.), ICMAS’96 Workshop on Norms, Obligations and Conventions, Kyoto, pp. 56-71. Revised version in In AI & Law, Vol.7, 1999, No.1, pp. 69-79.
15. Hilpinen (ed.) (1971). Deontic Logic: Introductory and Systematic Readings. Reidel.
16. Hogg, L., and Jennings, N.R. (2000). Variable sociability in agent-based decision making. In: N.R. Jennings, Y. Lesperance (eds.), Intelligent Agents VI. Proc. of the Sixth International Workshop on Agent Theories, Architectures and Languages, ATAL'99. Lecture Notes in AI, Springer Verlag. This volume.
17. Jennings. N.R., (1993). Commitments and conventions: The foundation of coordination in multi-agent systems. The Knowledge Engineering Review, 3, 223-250.
18. Jennings. N.R. and Campos, J.R. (1997). Towards a Social Level Characterisation of Socially Responsible Agents. IEEE Proc. on Software Engineering, vol. 144, 1, pp.11-25
19. Meyer, J.-J., Ch., and Wieringa, R.J. (eds.) (1993). Deontic Logic in Computer Science. John Wiley and Sons Ltd., Chicester, UK.
20. Rao, A.S., and Georgeff, M.P. (1991). Modeling rational agents within a BDI architecture. In: R. Fikes and E. Sandewall (eds.), Proc. of the Second Conference on Knowledge Representation and Reasoning, Morgan Kaufman, pp. 473-484.
21. Rosenschein, J., and Zlotkin, G. (1994). Rules of Encounters. MIT Press, Cambrige, USA.
22. Shoham, Y and Tenneholtz, M. (1992a). On the synthesis of useful social laws for artificial agent societies (preliminary report). Proceedings of the AAAI Conference, 276-281.

23. Shoham, Y. and Tenneholtz, M. (1992b). Emergent conventions in multi-agent systems: Initial experimental results and observations. Proceedings of the 3rd International Conference on KR&R. Cambridge, MA, 225-232.
24. Wagner, T., and Lesser, V. (2000). Relating quantified motivations for organizationally situated agents. In: N.R. Jennings, Y. Lesperance (eds.), Intelligent Agents VI. Proc. of the Sixth International Workshop on Agent Theories, Architectures and Languages, ATAL'99. LN AI, Springer Verlag. This volume.
25. Wright, G. von (1951), Deontic Logic. Mind, vol. 60, pp. 58-74.
26. Xuan, P., and V. Lesser, V., Incorporating uncertainty in agent commitments. In: N.R. Jennings, Y. Lesperance (eds.), Intelligent Agents VI. Proc. of the Sixth International Workshop on Agent Theories, Architectures and Languages, ATAL'99. Lecture Notes in AI, Springer Verlag. This volume.