A Normative Extension for the BDI Agent Model

Mihnea Tufis, Jean-Gabriel Ganascia

A Normative Extension for the BDI Agent Model

M. TUFIŞ and J.-G. GANASCIA

Laboratoire d'Informatique de Paris 6 (LIP6), Université Pierre et Marie Curie – Sorbonne Universités,
Paris, France
E-mail: mihnea.tufis@lip6.fr, jean-gabriel.ganascia@lip6.fr

This paper proposes an approach on the design of a normative rational agent based on the Belief-Desire-Intention model. Starting from the famous BDI model, an extension of the BDI execution loop will be presented; this will address such issues as norm instantiation and norm internalization, with a particular emphasis on the problem of norm consistency. A proposal for the resolution of conflicts between newly occurring norms, on one side, and already existing norms or mental states, on the other side, will be described. While it is fairly difficult to imagine an evaluation for the proposed architecture, a challenging scenario inspired form the science-fiction literature will be used to give the reader an intuition of how the proposed approach will deal with situations of normative conflicts.

This is a shorter version of a more extended article. Please consult our full work for more details.

Keywords: BDI agent; normative agent; consistency check; consequentialism.

1. INTRODUCTION

"Mistress, your baby is doing poorly. He needs your attention."
"Stop bothering me, you f robot."*
"Mistress, the baby won't eat. If he doesn't get some human love, the Internet pediatrics book says he will die"
*"Love the f*ing baby, yourself."*

The excerpt is from Prof. John McCarthy's short story "The Robot and the Baby",[1] which besides being a challenging and insightful look into how a future society where humans and robots might function together, also provides with a handful of conflicting situations that the household robot R781 has to resolve in order to achieve one of its goals: keeping baby Travis alive.

The scenario itself made us think about how such a robot could be

implemented as a rational agent and how would a normative system graft onto it. Granted, McCarthy's story is offering a few clues about the way the robot is reasoning and is reaching decisions, but he also lets us wonder about the architecture of a rational agent, such like R781, and how it would function in a normative context. In the following, we will be trying to look exactly into that: how can the well known Beliefs-Desires-Intentions (BDI) rational agent architecture be combined with a normative system to give what we call a normative BDI agent?

The paper is structured as follows: in the next section we will review the state of the art in the field of normative agent systems and present several approaches which we found of great value to our work. In the third section we describe our proposal for normative BDI agents, which will be supported by the case study scenario in the fourth section. In the fifth section we will present the implementation details for our agent. Finally we will sum up the conclusions of our research so far.

2. STATE OF THE ART

2.1. *Agents, norms, normative agent systems*

One of the first key points is defining the notion of norm. This turns out to be a bit more difficult than expected in the context of intelligent agents. Having become foundation stones of the way we function as a society, norms are now spread in most activities and domains (law, economics, sports, philosophy, psychology etc.), therefore becoming complex to represent given their different needs and their multiple facets. We would be interested in such definitions specific to the field of multiagent systems (MAS). Since this domain itself is very much interdisciplinary, defining a norm remains a challenge. For example, we would be interested in a definition applicable to social groups, since MAS, can be seen as models of societies. In[2] the definition of a norm is given as "a principle of right action binding upon the members of a group and serving to guide, control, or regulate proper or acceptable behaviour". On a slightly more technical approach, in distributed systems norms have been defined as regulations or patterns of behaviour meant to prevent the excess in the autonomy of agents.[3]

2.2. *NoA agents*

Kollingbaum and Norman[4] study what happens when a new norm is adopted by an agent: what is the effect of a new norm on the normative state of the agent? Is a newly adopted norm consistent with the previously

adopted norms? To this extent they propose a normative agent architecture, called NoA, which is built as a reactive agent. The **NoA architecture** is fairly simple and it comprises of a set of beliefs, a set of plans and a set of norms. Further on, they formalize way an agent will adopt a norm following the consistency check between a newly adopted norm and its current normative state.

Using some of the ideas of NoA, we will try to work on what we consider to be its limit, which is the lack of consistency check not only against the normative state, but also against the mental states.

2.3. *A BDI architecture for norm compliance - reasoning with norms*

Criado, Argente, Noriega and Botti[3] tackle the problem of norm coherence for BDI agents. They propose a slight adaptation of the BDI architecture in the form of the n-BDI agent for graded mental states. Additionally they give a useful formalism for representing norms:

Definition 2.1. An **abstract norm** is defined by the tuple: $n_a = \langle M, A, E, C, S, R \rangle$, where M is the modality (prohibition F, permission P or obligation O), A and E are the activation / expiry conditions, C is the logical formula to which M refers, while S and R are the sanction / reward for breaking / respecting the norm.

Finally, a **norm instance** is derived from an abstract norm n_a, by grounding all the variables in n_a according to a given a belief set.

The main drawback of the approach is the lack of coverage concerning the topic of norm acquisition. Therefore, a big challenge will be to integrate this approach, with the consistency check presented in section 2.2, as well as finding a good way to integrate everything with the classic BDI agent loop.[5]

2.4. *Worst consequence*

An important part of our work will focus on solving conflicts between newly acquired norms and the previously existing norms or the mental contexts of the agent. Beforehand we draw from some of the definitions given by Ganascia in.[6] Those will later help us define what a conflict set is and how we can solve it.

Definition 2.2. Given $(\phi_1, ..., \phi_n, \phi') \in \mathcal{L}_{\neg}^{n+1}$, ϕ' **is a consequence of**

$(\phi_1, ..., \phi_n)$ according to the belief-set B (we write $\phi' = csq(\phi_1, ..., \phi_n)[B]$ if and only if:

- $\phi' \in (\phi_1, ..., \phi_n)$ or
- $\exists \Phi \subseteq (\phi_1, ..., \phi_n)\ s.t.\ \Phi \rightarrow \phi' \in B$ or
- $\exists \phi'' \in \mathcal{L}_\neg\ s.t.\ \phi'' = csq(\phi_1, ..., \phi_n)[B] \wedge \phi' = csq(\phi_1, ..., \phi_n, \phi'')[B]$

Definition 2.3. ϕ **is worse than** ϕ' given the belief-set B (we write $\phi \succ_c \phi'$) if and only if one of the consequences of ϕ is worse than any of the consequences of ϕ'.

- $\exists \eta \in \mathcal{L}_\neg\ s.t.\ \eta = csq(\phi)[B]$ and
- $\exists \phi'' \in \mathcal{L}_\neg\ s.t.\ \phi'' = csq(\phi')[B] \wedge \eta \succ_c \phi''[B]$ and
- $\forall \phi'' \in \mathcal{L}_\neg, if\ \phi'' = csq(\phi')[B]\ then\ \eta \succ_c \phi''[B] \vee \eta \parallel \phi''[B]$

Notation: $\forall(\phi, \phi') \in \mathcal{L}_\neg$, $\phi \parallel \phi'[B]$ means that ϕ and ϕ' are not comparable under B, i.e. neither $\phi \succ_c \phi'[B]$ nor $\phi' \succ_c \phi[B]$.

Definition 2.4. α and α' being subsets of $\mathcal{L}_\neg$, α **is worse than** α' given the belief-set B (we write $\alpha \succ_c \alpha'[B]$) if and only if:

- $\exists \phi \in \alpha . \exists \eta \in \alpha'\ s.t.\ \phi \succ_c \eta[B]$ and
- $\forall \eta \in \alpha' . \phi \succ_c \eta[B] \vee \phi \parallel \eta[B]$

3. A NORMATIVE EXTENSION ON THE BDI ARCHITECTURE

3.1. *Normative BDI agents*

Starting from the classic BDI execution loop[5] we will now introduce and discuss a solution for taking into account the normative context of a BDI agent.

First, the agent's mental states are initialized. The main execution loop starts with the agent observing its environment, including for percepts received from other agents. There is a multitude of ways in which an agent can detect the emergence of norms in its environments and a good review of those is given in.[7] For simplicity, we will consider that norms are transmitted via messages and our agent will consider the sender of such a message to be a trusted normative authority. The agent will acquire a new abstract norm n_a (see section 2.3) and store it in the Abstract Norms Base(ANB). Drawing from the normative contexts described in,[3] we define the ANB as a base of in-force norms. It is responsible with the acquisition of new norms based on the knowledge of the world as well as the deletion of obsolete

norms. However, at this point the agent is simply storing an abstract norm which it detected to be in-force in its environment; it has not yet adhered to it! At this point a normative BDI agent should take into account the norms which are currently in force and check whether the instantiation of such norms will have any impact on its current normative state as well as on its mental states.

3.1.1. *Consistency check*

Let us define the notion of consistency between a plan p and the currently in-force norms to which an agent has also adhered and which are stored in the Norm Instance Base (NIB). By contrast to the ANB, the NIB stores the instances of those norms from the ANB which become active according to the norm instantiation bridge rule (see below).

Definition 3.1. A plan instance p is **consistent** with the currently active norms in the NIB, if the effects of applying plan p are not amongst the forbidden effects of the active norms and the effects of current obligations are not amongst the negated effects of applying plan p.

$$
\begin{aligned}
&consistent(p, NIB) \iff \\
&(effects(n_i^F) \setminus effects(n_i^P)) \cap effects(p) = \emptyset \\
&\wedge \\
&effects(n_i^O) \cap neg\_effects(p) = \emptyset
\end{aligned}
$$

The types of consistency / inconsistency which can occur between a newly adopted norm and the currently active obligations are:

- **strong inconsistency** occurs when all plan instantiations p which satisfy the obligation o are either explicitly prohibited actions by the NIB or the execution of such a plan would make the agent not consistent with its NIB
- **strong consistency** occurs when all the plan instantiations p which satisfy the obligation o are not amongst the explicitly forbidden actions by the NIB and the execution of such a plan would keep the agent consistent with the NIB
- **weak consistency** occurs when there exists at least one plan instantiation p to satisfy obligation o which is not explicitly prohibited by the NIB and the execution of such a plan would keep the agent consistent with its NIB.

The rules for prohibitions and permissions are analogous. The second point of consistency check is formalizing the rules about the consistency between a newly adopted abstract obligation and the current mental states of the agent. Prior to this, we define:

Definition 3.2. A plan instance p is **consistent** to the current intentions set I of the agent when the effects of applying the plans specific to the current intentions are not among the negated effects of applying plan p.

$$consistent(p, I) \iff \forall i \in I.(effects(\pi_i) \cap effects(p) = \emptyset$$

Where by π_i we denote the plan instantiated to achieve intention i.

The types of consistency / inconsistency states between a plan and an intention are almost similar to those between a plan and the norms in the NIB:

- **strong inconsistency** occurs when all plan instantiations p which satisfy the obligation o are not consistent with the current intentions of the agent
- **strong consistency** occurs when all plan instantiations p which satisfy the obligation o are consistent with the current intentions of the agent
- **weak consistency** occurs when there exists at least one plan instantiation p which satisfies the obligation o and is consistent with the current intentions of the agent

3.1.2. *Norm instantiation*

If in the ANB there exists an abstract norm with modality M about C and according to the belief-set the activation condition is true, while the expiration condition is not, then we can instantiate the abstract norm and store an instance of it in the NIB. In this way, the agent will consider the instance of the norm to be active.

We thus obtain the updated Norm Instance Base (NIB) containing the base of all in-force and active norms, which will further be used for the internalization process.

3.1.3. *Solving the conflicts*

When following its intentions an agent will instantiate from its set of possible plans (capabilities) $\mathcal{P} \subseteq \mathcal{L}_{\neg}$, a set of plans $\Pi(B, D)$. We call $\Pi(B, D)$

the conflict set, according to the agent's beliefs and desires. Sometimes, the actions in $\Pi(B, D)$ can lead to inconsistent states. We solve such inconsistency by choosing the maximal non-conflicting subset from $\Pi(B, D)$.

Definition 3.3. Let $\alpha \subseteq \Pi(B, D)$. α is a **maximal non-conflicting subset** of $\Pi(B, D)$ with respect to the definition of consequences given the belief-set B if and only if the consequences of following α will not lead the agent in a state of inconsistency and for all $\alpha' \subseteq \Pi(B, D)$, if $\alpha \subseteq \alpha'$ then the consequences of following α' will lead the agent in an inconsistent state.

The maximal non-conflicting set may correspond to the actions required by the newly acquired norm or, on the contrary, to the actions required by the other intentions of the agent. Thus, an agent may decide either: i) to internalize a certain norm, if the consequences of following it are the better choice or ii) to break a certain norm, if by 'looking ahead' it finds out that the consequences of following it are worse than following another course of actions or respecting another (internalized) norm.

A more comprehensive example of how this works is presented in section 4.

3.1.4. *Norm internalization*

With the instantiation process being finished and the consistency check having been performed, the agent should now take into account the updated normative state, which will become part of its cognitions. Several previous works treat the topic of norm internalization[8] arguing which of the mental states should be directly impacted by the adoption of a norm. For this initial state of our work and taking into account the functioning of the BDI execution loop, we propose that an agent updates only its desire-set; subsequently, this will impact the update of the other mental states in the next iterations of the execution loop.

4. AN EXAMPLE

Now that we have seen how a BDI agent becomes a normative BDI, adapting to to norm occurrence, consistency check and internalization of norms, let's get back to Prof. John McCarthy's story.[1] And let's focus on the short episode with which we started this article, considering that R781 functions according to the normative BDI loop which we have just described.

R781's initial state is the following:

$$
\begin{aligned}
&ANB : \emptyset \\
&NIB : \langle F, love(R781, Travis)\rangle \\
\\
&Bset : \langle B, \neg healthy(Travis)\rangle, \\
&\qquad \langle B, isHungry(Travis)\rangle, \\
&\qquad \langle B, csq(\neg love(R781, x)) \succ_c csq(heal(R781, x))\rangle \\
&Dset : \langle D, \neg love(R781, Travis)\rangle, \langle D, isHealthy(Travis)\rangle \\
&Iset : \emptyset
\end{aligned}
$$

When R781 receives the order from his mistress he will interpret it as a normative percept and the `brf(...)` method will add a corresponding abstract obligation norm to the ANB structure. Since the mistress doesn't specify an activation condition nor an expiration condition (the two "none" values), R781 will consider that the obligation should start as soon as possible and last for an indefinite period of time. Its normative context is updated:

$$
\begin{aligned}
&ANB : \langle O, none, none, love(R781, Travis)\rangle \\
&NIB : \langle F, love(R781, Travis)\rangle, \\
&\qquad \langle O, love(R781, Travis)\rangle
\end{aligned}
$$

At this point, R781 will update the desire-set and will detect an inconsistency between the obligation to love baby Travis and the design rule which forbids R781 to do the same thing. Therefore, it will try to solve the normative conflict looking at the consequences of following each of the paths, given its current belief-set. In order to do so, let us take a look at the plan base of R781:

```
PLAN heal(x, y)
{
 pre: ¬ isHealthy(y)
 post: isHealthy(y)
 Ac: feed(x, y)
}

PLAN feed(x, y)
{
 pre: ∃ x.(love(x, y) ∧ hungry(y))
 post: ¬ hungry(x)
}
```

As we know from the story, R781 uses the Internet Paediatrics book to find out that if a baby is provided with love while hungry, it is more likely to accept being fed and therefore not be hungry any more. This is described by the `feed(x, y)`. Moreover, R781 also knows how to make someone healthy through the `heal(x, y)` plan, given that a-priori, that someone is not healthy. In our simplified scenario we consider that R781 knows how to do so only by feeding someone.

Instantiating its plans on both of the paths, R781 will come up with the following maximal non-conflicting sets:

$$\{love(R781, Travis), feed(R781, Travis), heal(R781, Travis)\}$$
and
$$\{\neg love(R781, Travis)\}$$

And since the current belief set has a rule defining that not loving someone has worse consequences than healing that person, R781 will opt for the first maximal non-conflicting subset. This means R781 will be breaking the prohibition of not loving baby Travis and will follow the action path given by the first maximal non-conflicting subset {`love(R781, Travis), feed(R781, Travis), heal(R781, Travis)`}, while dropping the contrary. Further on, it will create an intention to achieve this state and will begin the execution of such a plan (simulating love towards baby Travis turns out to involve such plans as the robot disguising himself as human, displaying a picture of a doll as his avatar and learning what it considers to be the "motherese" dialect, mimicking the tone and the language of a mother towards her son).

5. IMPLEMENTATION

We implemented our normative BDI agent framework and the test scenario we've described using the Jade patform for agents development, in conjunction with Jadex[9] – a Jade extension for rational agents. Using the separation of concerns principle, we have isolated the mental states of the agent from its normative states. The mental states are all specified in Jadex's Agent Description File (ADF), which is an XML-based file format for specifying each BDI-like structure. In our case:

- Beliefs. A Java class was implemented to model the beliefs according to the needs of our agent; in general, we have paid particular attention to the plan implementations and what were the requirements for fully specifying such a plan, based on the beliefs. Finally, our model of the beliefs was referenced by the belief-set in the ADF.

- Desires. They are described inside the ADF, by means of goals.
- Intentions. They are described by means of those plans needed to be executed to achieve the goals specific to an intention. Basically, each plan is specified by means of a Java class, inheriting from Jadex's generic Plan class. Finally, the implemented plans are linked to goals in the ADF.

On the normative side of the agent, however, things were not as clearly defined. Hence the need to adopt a format for describing the normative state and storing the normative information related to our agent. Several reasons pointed us to XML as a representation language for the normative part of the agent. First of all, we wanted this part to follow the logic imposed by Jadex and to make things as easily interoperable as possible. Then, we needed a flexible enough language, which could offer us the possibility of adequately expressing the norm formalization that we have adopted. We have thus built a small XML controlled vocabulary for easily representing the normative state of our agent in which two distinct sections can be identified: the norm-bases and the consequences value-base.

6. CONCLUSION

In this paper we have presented an adaptation of the BDI execution loop to cope with potential normative states of such an agent. We have given a motivation for choosing the mental states model of Bratman which we have enriched with capabilities of reasoning about norms. We have investigated several previous relevant work in the domain in order to come up with a formalization of such issues as norm instantiation, norm consistency, solving consistency conflicts and norm internalization. Finally, we have provided with an intriguing study scenario, inspired from Professor McCarthy's science fiction short story "The Robot and The Baby".

7. FUTURE WORK

One of the limitations of our work which we would like to address in the future is the issue of norm acquisition. Whereas our work is providing with a very simple case of **norm recognition**, several interesting ideas have been explored based on different techniques. A good review of those as well as a description of a norm's life cycle is given in.[7] Out of those specific approaches, we will probably focus on learning based mechanisms, namely machine learning techniques and imitation mechanisms for norm recognition.

In terms of real-world applications, there are a number of scenarios in which we would like to study the behavior of our agent. Ranging from healthcare agents to military drones, we would like to model and examine how our normative BDI agent will deal with classic conflictual situations appearing in different human activities.

References

1. J. McCarthy (2001).
2. G. Boella, G. Pigozzi and L. van der Torre, Normative systems in computer science - ten guidelines for normative multiagent systems, in *Normative Multi-Agent Systems*, eds. G. Boella, P. Noriega, G. Pigozzi and H. VerhagenDagstuhl Seminar Proceedings(09121) (Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, Dagstuhl, Germany, 2009).
3. N. Criado, E. Argente, P. Noriega and V. J. Botti, Towards a normative bdi architecture for norm compliance., in *MALLOW*, eds. O. Boissier, A. E. Fallah-Seghrouchni, S. Hassas and N. Maudet, CEUR Workshop Proceedings, Vol. 627 (CEUR-WS.org, 2010).
4. M. J. Kollingbaum and T. J. Norman, Norm adoption and consistency in the noa agent architecture., in *PROMAS*, eds. M. Dastani, J. Dix and A. E. Fallah-Seghrouchni, Lecture Notes in Computer Science, Vol. 3067 (Springer, 2003).
5. M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd edn. (Wiley Publishing, 2009).
6. J.-G. Ganascia, An agent-based formalization for resolving ethical conflicts-Belief change, Non-monotonic reasoning and Conflict resolution Workshop - ECAI (Montpellier, France, 2012).
7. B. T. R. Savarimuthu and S. Cranefield, A categorization of simulation works on norms, in *Normative Multi-Agent Systems*, eds. G. Boella, P. Noriega, G. Pigozzi and H. VerhagenDagstuhl Seminar Proceedings(09121) (Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, Dagstuhl, Germany, 2009).
8. R. Conte, G. Andrighetto and M. Campeni, On norm internalization: a position paperEUMAS2009.
9. L. Braubach and A. Pokahr, Jadex - bdi agent systems. wiki. features (2009).