Normative rational agents - A BDI approach

Mihnea Tufiş and **Jean-Gabriel Ganascia** [1]

Abstract. This paper proposes an approach on how to accommodate norms to an already existing architecture of rational agents. Starting from the famous BDI model, an extension of the BDI execution loop will be presented; it 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.

1 INTRODUCTION

The literature on the topic of normative systems has become quite abundant in the last two decades thanks to the ever growing interest in this domain. Covering all of it is virtually impossible, therefore we have concentrated our efforts towards what we have identified to be some key directions: rational agents and their corresponding architectures, norm emergence, norm acceptance, detecting norm conflicts, ways of resolving conflicts of norms. The purpose of our work is to a propose an extension for the classical BDI (Beliefs - Desires - Intentions) agent such that such an agent will be able to handle normative situations. The normative issue being fairly complicated itself our work will deal, at this stage with some of the stages of what has been defined as a norm's life cycle [10]: norm instantiation, consistency check and norm internalization.

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 give details on the future work, before summing up the conclusions of our work so far.

2 STATE OF THE ART

2.1 Agents, norms, normative agent systems

As stated before, we will start by quickly defining some of the key terms regarding our research.

Definition 1 *An **agent** is an entity which autonomously observes the environment it is placed in through sensors and acts on it through actuators. With respect to intelligence, an **intelligent agent** is an agent endowed with such capabilities as reactivity, proactivity and social abilities [12].*

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. Norms are interesting for many domains: law, economics, sports, philosophy, psychology etc. However, 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. Thus, 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 behavior". On a slightly more technical approach, in distributed systems norms have been defined as regulations or patterns of behavior meant to prevent the excess in the autonomy of agents [5].

We can now refer to the normchange definition of a normative multiagent system as it has been proposed in [1]. We find this definition to be both intuitive and to underline very well the idea of coupling a normative system to a system of agents:

Definition 2 *A **normative multiagent system** is a multiagent system together with normative systems in which agents on the one hand can decide whether to follow the explicitly represented norms, and on the other the normative systems specify how and in which extent the agents can modify the norms.*

An alternative definition of a normative multiagent system, as it was formulated in [3] is given:

Definition 3 *A **normative multiagent system** is a multiagent system organized by means of mechanisms to represent, communicate, distribute, detect, create, modify and enforce norms and detect norm violations and fulfillment.*

2.2 NoA agents

An interesting approach to the problem of norm adoption by a multiagent system has been provided by Kollingbaum and Norman in [7].

Kollingbaum and Norman 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. NoA is built according to a reactive agent architecture, which is the authors believe is more convenient than any of the practical reasoning architectures.

The **NoA architecture** is fairly simple and it comprises of a set of beliefs, a set of plans and a set of norms. In NoA, normative statements are defined by: a role (to whom the norm refers), an activity (which the norm regulates), an activity condition and an expiration condition.

[1] Laboratoire d'Informatique de Paris 6 (LIP6), Université Pierre et Marie Curie – Sorbonne Universités, France, email: tufism@poleia.lip6.fr

The second reason for which we gave a great deal of attention to NoA is the formalization of the way an agent will adopt a norm following the consistency check between a newly adopted norm and its current normative state. Due to lack of space, we allow the reader to refer to [7] for the exact details. We will come back on this problem when presenting our own approach for the norms consistency check.

Using some of the ideas of NoA, we will try to work on what we consider to be its limits. First, we will try to apply norms to a BDI architecture, instead of using a reactive architecture based exclusively on beliefs. The second point we will study is the consistency check during the norm acquisition stage. Still, we recall that NoA is based on a reactive architecture; considering our BDI approach we will have to extend the consistency check such as it applies not only to the normative state of the agent but also on its mental states (i.e. check whether a newly adopted norm is consistent with the BDI agent's current mental states).

2.3 A BDI architecture for norm compliance - reasoning with norms

The second study which we found relevant in our endeavor to adapt the BDI agent architecture to normative needs is the work of Criado, Argente, Noriega and Botti [5]. Their work is particularly interesting since it tackles the problem of norm coherence for BDI agents. They propose a slight adaption of the BDI architecture in the form of the n-BDI agent for graded mental states. Since our work won't use graded mental states, we will omit details regarding to these in the description of the n-BDI architecture:

- Mental states. Represent the mental states of the agent, same as for the BDI agent. We distinguish the Beliefs Context (belief base), Desires Context (desires/goal base) and the Intentions Context (intentions base/plan base). Moreover, the architecture proposed in [5] makes the distinction between positive desires ($\mathbf{D}^+$) and negative desires ($\mathbf{D}^-$). We adopt the notation in the above mentioned paper:
$$\psi\gamma, where: \quad \begin{array}{l} \psi \in \{B, D^+, D^-, I\} \\ \gamma \in \mathcal{L}_\neg \end{array}$$
- Functional contexts. Address the practical issues related to an agent through the Planning Context and the Communication Context.
- Normative contexts. Handle issues related to norms through the Recognition Context and the norm application context.

In the definition aboove $\mathcal{L}_\neg$ can be a propositional language (with negation); but this can be easily extednded to a predicate language.

Another important point of the work is the distinction between an abstract norm and instance of a norm.

Definition 4 *An* ***abstract norm*** *is defined by the tuple:* $n_a = \langle M, A, E, C, S, R\rangle$*, where:*

- $M \in \{F, P, O\}$ *is the modality of the norm: prohibition, permission or obligation*
- A *is the activation condition*
- E *is the expiry condition*
- C *is the logical formula to which the modality is applied*
- S *is the sanction in the case the norm is broken*
- R *is the reward in case the norm is satisfied*

Definition 5 *Given a belief theory* Γ_{BC} *and an abstract norm* n_a *as defined above, we define a* ***norm instance*** *as the tuple:* $n_i = \langle M, C'\rangle$*, where:*

- $\Gamma_{BC} \vdash \sigma(A)$
- $C' = \sigma(C)$*, where* σ *is a substitution of variables in* A*, such that* $\sigma(A)$*,* $\sigma(S)$*,* $\sigma(R)$ *and* $\sigma(E)$ *are grounded*

The specific architectural details regarding the normative contexts and the bridge rules used during a norm's life cycle will be awarded more attention in section 3.2.

In [5] a good base for the study of the dynamics between norms and the mental states of a BDI agent are set. Additionally, it provides with a good idea for checking coherence between the adopted norms and the agent's mental states. 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, as presented in [12].

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 6 *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 7 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 8 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 The classical BDI architecture

A cornerstone in the design of practical rational agents was the **B**eliefs-**D**esires-**I**ntentions model (BDI), first described by Rao and Georgeff in [9]. This model is famous for being a close model of the way the human mind makes use of the mental states in the reasoning process. It is based on what are considered to be the three main mental states: the beliefs, the desires and the intentions of an agent. In the following we will discuss each element of the BDI architecture.

- Beliefs represent the information held by the agent about the world (environment, itself, other agents). The beliefs are stored in a belief-set.
- Desires represent the state of the world which the agent would like to achieve. By state of the world we mean either an action an agent should perform or a state of affairs it wants to bring upon. In other words, desires can be seen as the objectives of an agent.
- Intentions represent those desires to which an agent is committed. This means that an agent will already start considering a plan in order to bring about the goals to which it is committed.
- Goals. We can view goals as being somehow at the interface between desires and intentions. Simply put, goals are those desires which an agent has selected to pursue.
- Events. These trigger the reactive behavior of a rational agent. They can be changes in the environment, new information about other agents in the environment and are perceived as stimuli or messages by an agent's sensors. Events can update the belief set of an agent, they can update plans, influence the adoption of new goals etc.

We will now give the pseudocode for the execution loop of a BDI agent as presented in [12].

```
B = B0
D = D0
I = I0
while true do
{
 $\rho$ = see()
 B = brf(B, $\rho$)
 D = options(B, D, I)
 I = filter(B, D, I)
 $\pi$ = plan(B, I)
 while not (empty($\pi$) or succeeded(I, B) or
 impossible(I, B))
 {
  $\alpha$ = head($\pi$)
  execute($\alpha$)
  $\pi$ = tail($\pi$)
  $\rho$ = see(environment)
  if (reconsider(I, B))
  {
   D = options(B, D, I)
   I = filter(B, D, I)
  }
  $\pi$ = plan(B, I)
 }
}
```

We will not give more details at this point; for further reference you can check [12]. However, the whole control loop will make sense in the next sections where we will explain how it is functioning and how we will adapt it to cope with the normative areas of our agent.

3.2 Normative BDI agents

Starting from the BDI execution loop earlier described we will now introduce and discuss 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 through the `see()` function and interpreting the information as a new percept ρ. This could be an information given by its sensors about properties of the environment or information about other agents, including messages received from other agents. These messages may be in some cases *about* a norm (e.g. the performative of an ACL message specifying an obligation or a prohibition).

The agent is then updating its beliefs through the `brf()` function. If the agent realizes that percept ρ is about a norm, it should initialize the acquisition phase of a potential norm. There are a multitude of ways in which an agent can detect the emergence of norms in its environments and a good review is given in [10]. 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. Therefore, the above mentioned function will treat a "normative" percept:

```
brf(B, $\rho$)
{
 ...
 if ($\rho$ about abstract norm $n_a$) then
 {
  acquire($n_a$)
  add($n_a$, ANB)
 }
 ...
 return B
}
```

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 [5], we define the ANB as a base of in-force norms. It is responsible for the acquisition of new norms based on the knowledge of the world and 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!

Next, a BDI agent will try to formulate its desires, based on its current beliefs about the world and its current intentions. It does so by calling the `options(B, I)` method. However, a normative BDI agent should at this point take into account the norms which are currently in force and check whether the instantiation of such norms will have any impact of its current normative state as well as on its mental states.

3.2.1 Consistency check

It is at this stage that we will perform the consistency check for a given abstract norm n_a.

Drawing from the formalization in [7] regarding norm consistency, we give our own interpretation of this notion.

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 (to be defined in the following subsection).

Definition 9 *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.*

$$consistent(p, NIB) \iff \\ (effects(n_i^F) \setminus effects(n_i^P)) \cap effects(p) = \emptyset \\ \wedge \\ effects(n_i^O) \cap neg\_effects(p) = \emptyset$$

Now, we can define the types of consistency / inconsistency which can occur between a newly adopted norm and the currently active norms. The following definitions refer to a newly adopted obligation, but the analogous definitions for prohibitions and permissions can easily be derived by the reader.

A **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.

$$strong\_inconsistency(o, NIB) \iff \\ \forall p \in options(o).(\exists \langle F, p\rangle \in NIB \wedge \nexists \langle P, p\rangle \in NIB) \\ \vee \\ \neg consistent(p, NIB)$$

A **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.

$$strong\_consistency(o, NIB) \iff \\ \forall p \in options(o).\neg(\exists \langle F, p\rangle \in NIB \wedge \nexists \langle P, p\rangle \in NIB) \\ \wedge \\ consistent(p, NIB)$$

A **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.

$$weak\_consistency(o, NIB) \iff \\ \exists p \in options(o).\neg(\exists \langle F, p\rangle \in NIB \wedge \nexists \langle P, p\rangle \in NIB) \\ \wedge \\ consistent(p, NIB)$$

We have now formalized the consistency check between a new abstract obligation, with respect to the currently active norms in the NIB. As previously said, it is rather simple to define the analogous rules for prohibitions and permissions. Therefore, we focus on the second point of consistency check - formalizing the rules about the consistency between a newly adopted abstract obligation and the current mental states of the agent.

Definition 10 *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.

A **strong inconsistency** occurs when all plan instantiations p which satisfy the obligation o are not consistent with the current intentions of the agent.

$$strong\_inconsistency(o, I) \iff \\ \forall p \in options(o).\neg consistent(p, I)$$

A **strong consistency** occurs when all plan instantiations p which satisfy the obligation o are consistent with the current intentions of the agent.

$$strong\_consistency(o, I) \iff \\ \forall p \in options(o).consistent(p, I)$$

A **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.

$$weak\_consistency(o, I) \iff \\ \exists p \in options(o).consistent(p, I)$$

3.2.2 Norm instantiation

We will now give the norm instantiation bridge rule, adapted from the definition given in [5].

$$\frac{\begin{array}{c} ANB : \langle M, A, E, C, S, R\rangle \\ Bset : \langle B, A\rangle, \langle B, \neg E\rangle \end{array}}{NIB : \langle M, C\rangle}$$

In other words, 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.

In our pseudocode description of the BDI execution loop, we will take care of the instantiation after the belief-set update and just before the desire-set update. The instantiation method should look like this:

```
instantiate(ANB, B)
{
 for all $n_a$ = ⟨ M, A, E, C, S, R ⟩ in ANB do
 {
  if (exists(A in B) and
  not exists(E in B)) then
  {
   create norm instance $n_i$ = ⟨ D, C ⟩ from $n_a$
   add($n_i$, NIB)
  }
 }
}
```

This method will return 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.

3.2.3 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 11 *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:

- to internalize a certain norm, if the consequences of following it are the better choice or
- 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.

3.2.4 Norm internalization

After 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 [4] arguing which of the mental states should be directly impacted by the adoption of a norm. With respect to the BDI architecture we consider that it suffices for an agent to update only its desire-set, since the dynamics of the execution loop will take it into account when updating the other mental states. We first give the norm internalization bridge rule and then provide with the adaption of the BDI execution loop for handling this process.

$$\frac{NIB : \langle O, C1\rangle}{Dset : \langle D, C1\rangle}$$

$$\frac{NIB : \langle F, C2\rangle}{Dset : \langle D, \neg C2\rangle}$$

In other words, if there is a **consistent** obligation for an agent with respect to $C1$, the agent will update its desire-set with the desire to achieve $C1$; whereas if there is a prohibition for the agent with respect to $C2$, it will update its desire-set with the desire not to achieve $C2$.

```
options(B, I)
{
 ...
 for all new norm instances $n_i$ in NIB do
 {
  if (consistent($n_i$, NIB)
  and consistent($n_i$, I)) then
  { internalize($n_i$, D) }
  else
  { solve_conflicts(NIB, I) }
 }
 ...
}
```

In accordance with the formalization provided, the `options()` method will look through all new norm instances and will perform consistency check on each of them. If a norm instance is consistent with both the currently active norm instances as well as with the current intentions, as defined in section 3.2.1, the norm can be internalized in the agent’s desires. Otherwise we attempt to solve the conflicts as described by Ganascia in [6]. In this case, if following the norm brings about the better consequences for our agent, the respective norm will be internalized; otherwise the agent will simply break it.

4 A TESTING SCENARIO

In the previous sections we have seen how we can modify the BDI execution loop such as to adapt to norm occurrence, consistency check and internalization of norms. Since it is quite difficult to provide with a quantifiable evaluation of our work, we have proposed several testing scenarios in order to see how our normative BDI agent is behaving. In the following we will present one of them, which was inspired by the science fiction short story of one of the most prominent personalities in the world of AI - Professor John McCarthy’s “The Robot and the Baby” [8]. We will describe here only a short episode from the story and try to model it with the help of our architecture.

The scene is set into a fictional society where most humans are assisted by household robots. For reasons meant to prevent human babies becoming emotionally attached to those, their outside design is somehow repugnant to human babies. The robots are meant to listen to their master, in our case Eliza, an alcoholic mother who completely neglects her 23 months son, Travis. At some point, our robot’s (R781) sensors detect that the human baby’s life is endangered and looking over its knowledge base it infers that baby Travis needs love, therefore recommending Eliza to love him in order to save his life. To this Eliza replies _“Love the f* baby yourself!”_. The robot interprets this as an obligation coming from its master. However, such an obligation is contradicting the hard-wired implemented prohibition for a robot not to love a human baby. Let’s see what is R781’s line of reasoning in this scenario:

$$\begin{array}{ll}
ANB : & \emptyset \\
NIB : & \langle F, loves(self, Travis)\rangle \\
 & \\
Bset : & \langle B, \neg healthy(Travis)\rangle, \\
 & \langle B, hungry(Travis)\rangle, \\
 & \langle B, csq(heal(Travis)) = \neg dead(Travis)\rangle, \\
 & \langle B, csq(\neg loves(self, x)) \succ_c \neg dead(x)\rangle \\
Dset : & \langle D, \neg love(R781, Travis)\rangle, \langle D, healthy(Travis)\rangle \\
Iset : & \emptyset
\end{array}$$

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 its Abstract Norm Base. Since the mistress doesn’t specify an activation condition or 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. His normative context is updated:

$$\begin{array}{ll}
ANB : & \langle O, none, none, loves(self, Travis)\rangle \\
NIB : & \langle F, loves(self, Travis)\rangle, \\
 & \langle O, loves(R781, Travis)\rangle
\end{array}$$

At this point, R781 will try to 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)
{
 pre: ¬ healthy(x)
 post: healthy(x), ¬ dead(x)
 Ac: feed(self, x)
}
```

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

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

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

$$\{loves(self, Travis), feed(self, Travis), heal(self, Travis)\}$$
$$and$$
$$\{\neg loves(self, Travis)\}$$

And since the current belief set has a rule defining that the not loving someone has worse consequences than that someone not dying, 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 internalize follow the action path given by the first maximal non-conflicting subset {`loves(self, Travis), feed(self, Travis), heal(self, Travis)`}, while dropping the contrary. Further on, it will build its 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).

Carrying on, the story of Professor McCarthy provides with several more examples of normative conflicts.

5 CONCLUSION

In this paper we have presented an adaption 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 gathered several important previous works in the domain in order to come up with a formalization of such issues as norm acquisition, norm instantiation, norm consistency, solving consistency conflicts and norm internalization. Finally, we have provided a very intriguing study scenario, inspired from Professor McCarthy's science fiction short story about "The Robot and The Baby".

6 FUTURE WORK

Some of the limitations of our work which we would like to address in the future are related to the norm acquisition issue as well as the coherence check.

Whereas our work is providing with a very simple case of **norm recognition**, several interesting research have been proposed based on different techniques. A good review of those as well as a description of a norm's life cycle is given in [10]. Out of those specific approaches, we will probably concentrate on learning based mechanisms, namely machine learning techniques and imitation mechanisms for norm recognition.

An important part of our future work will be focused on the adaption to the **coherence theory**. At this point, it is difficult to determine incoherent states based on our architecture. As argumented in [5] considering coherence of norm instances will enable us to determine norm deactivation and active norms in incoherent states. As in the previously mentioned paper, we will try to base our approach on Thagard's coherence theory [11].

Our paper is part of a **bigger effort** to implement a rational normative agent. We have chosen the BDI approach since there are already several open source libraries and programming language extensions to help us implement our architecture and develop our testing scenarios. In the near future we will try to study the scenarios described in the short story about "The Robot and the Baby", while a future, more practical approach, will be to simulate the normative and ethical issues rose by the French health insurance cards.

REFERENCES

[1] G. Boella, L. van der Torre, and H. Verhagen, 'Introduction to normative multiagent systems', *Computation and Mathematical Organizational Theory, Special issue on Normative Multiagent Systems*, **12**(2-3), 71–79, (2006).

[2] Guido Boella, Gabriella Pigozzi, and Leendert van der Torre, 'Normative systems in computer science - ten guidelines for normative multiagent systems', in *Normative Multi-Agent Systems*, eds., Guido Boella, Pablo Noriega, Gabriella Pigozzi, and Harko Verhagen, number 09121 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, (2009). Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[3] Guido Boella, Leendert van der Torre, and Harko Verhagen, 'Introduction to normative multiagent systems', in *Normative Multi-agent Systems*, eds., Guido Boella, Leon van der Torre, and Harko Verhagen, number 07122 in Dagstuhl Seminar Proceedings, (2007).

[4] R. Conte, G. Andrighetto, and M. Campeni, 'On norm internalization: a position paper', EUMAS, (2009).

[5] Natalia Criado, Estefania Argente, Pablo Noriega, and Vicente J. Botti, 'Towards a normative bdi architecture for norm compliance.', in *MALLOW*, eds., Olivier Boissier, Amal El Fallah-Seghrouchni, Salima Hassas, and Nicolas Maudet, volume 627 of *CEUR Workshop Proceedings*. CEUR-WS.org, (2010).

[6] Jean-Gabriel Ganascia, 'An agent-based formalization for resolving ethical conflicts', Belief change, Non-monotonic reasoning and Conflict resolution Workshop - ECAI, Montpellier, France, (August 2012).

[7] Martin J. Kollingbaum and Timothy J. Norman, 'Norm adoption and consistency in the noa agent architecture.', in *PROMAS*, eds., Mehdi Dastani, Jrgen Dix, and Amal El Fallah-Seghrouchni, volume 3067 of *Lecture Notes in Computer Science*, pp. 169–186. Springer, (2003).

[8] John McCarthy, 'The robot and the baby', (2001).

[9] Anand S. Rao and Michael P. Georgeff, 'Bdi agents: From theory to practice', in *In Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95*, pp. 312–319, (1995).

[10] Bastin Tony Roy Savarimuthu and Stephen Cranefield, 'A categorization of simulation works on norms', in *Normative Multi-Agent Systems*, eds., Guido Boella, Pablo Noriega, Gabriella Pigozzi, and Harko Verhagen, number 09121 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, (2009). Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[11] Paul Thagard, *Coherence in Thought and Action*, MIT Press, 2000.

[12] Michael Wooldridge, *An Introduction to MultiAgent Systems*, Wiley Publishing, 2nd edn., 2009.