Identifying Conflicts between Norms and Values

Karen da Silva Figueiredo and Viviane Torres da Silva

Computer Science Department, Universidade Federal Fluminense (UFF),
Rua Passos da Pátria 156, Bloco E, 24210-240, Niterói, Brazil
{kfigueiredo, viviane.silva}@ic.uff.br

Abstract. Norms and values are part of the organizational culture. While the values support the agent's autonomy by representing its character and helping to make decisions and execute actions, the norms are used by the system to cope with the autonomy of the agents by regulating their behavior and the execution of actions. One of the main challenges faced by agents at runtime is the conflicts that may arise between the systems norms and their values. The goals of this paper are to point out the conflict cases between norms and values and to propose an algorithm to help the agent to identify those conflict cases.

Keywords: Norm, Value, Conflict, Identification, Algorithm.

1 Introduction

Culture can be seen as a social accepted collection of behavior patterns learned, shared and transmitted with the aim of adapt the individuals of a group [1, 2]. These behavior patterns are expressed by the group in many forms of cultural mechanisms, such as norms and values.

Norms are used to regulate the behavior of the agents by describing the actions they must (or must not) execute in specific situations in order to cope with the heterogeneity, autonomy and diversity of interests of those agents. On the other hand, values are used to help the agents to evaluate and decide what to do based on what they believe is right to themselves and to their group [3]. The actions executed by the agents will promote or demote their values. Thus, the agents will always prefer to execute the actions that promote more values (or demote fewer values).

Since norms are used to regulate the actions of an agent and its values are promoted or demoted due to execution of such actions, conflicts can arise between the norms and its values. For instance, a norm can prohibit the agent to execute an action that promotes one of its values. Although there are several works proposing solutions for the checking of conflicts between norms [4-8] and between norms and some agents’ elements, such as goals [9], there is a need for an approach to check for conflicts between norms and values. In this paper we point out the conflict cases that may arise between norms and values and propose an algorithm to identify those conflict cases.

As far as we know, this is the first MAS work that investigates the identification of this kind of conflicts. The identification of the conflicts between the system's norms
and the agents values is an important issue, because by identifying and solving the conflicts the agents are able to decide, for instance: (i) whether or not enter an organization once they realize if the organization's norms are coherent or not with their values; and (ii) which actions they will execute and, as consequence, which norms they will fulfill or violate.

The reminder of this paper is organized as follows. In Section 2 we provide some background material about norms (Section 2.1), values (Section 2.2) and the specification notation adopted (Section 2.3) for the understanding of this work. Section 3 is the core of this paper, where we discuss the norms and values relation, address the conflict cases between them (Section 3.1), present the proposed algorithm to identify these conflict cases (Section 3.2) and show an example (Section 3.3). To finish, Section 4 concludes the paper with final remarks and discusses future work.

2 Background

In this section some background material for the understanding of this work is presented. We introduce the basic concepts about norms and values in Sections 2.1 and 2.2, respectively, in order to support the conflicts discussion of Section 3. In Section 2.3 we summarize the Z language that was used to write the algorithm to identify conflicts between norms and values proposed in this work.

2.1 Norms

Norms are mechanisms used to regulate the behavior of the agents by describing the actions that can be performed (permissions), actions that must be performed (obligations), and actions that cannot be performed (prohibitions). They represent a way for agents to understand their responsibilities and the responsibilities of the other agents in the system.

A norm can be described by several elements, the main elements that compose a norm are [4]: the deontic concept (i.e. obligation, prohibition and permission) that designate the behavior restrictions for the agents; the involved entities who have their behavior restricted by the norm; the actions being regulated by the norm; the activation constraints that limit the period during while the norm is active; the sanctions to be applied when the norm is violated or fulfilled; and the context that determines the area of application of the norm.

Due to the dynamics of MAS, conflicts may arise between the system norms and other system elements as goals, e.g. [9], values and even other norms, e.g. [4-8]. The agents must be able to cope with these conflicts, identifying and solving them at running time in order to make decisions, select intentions and execute actions. In this work we focus on the identification of conflicts between the system’s norms and the agent’s values (see Section 3).
2.2 Values

In this work we address the definition of value of the Schwartz Value Theory (SVT) [10] because it is the state-of-the-art theory about value [11] and it has been previously adopted in MAS research [12]. Values are described by SVT as concepts about desirable end states or behaviors that transcend specific situations used to guide the selection or evaluation of behavior and events that are ordered by relative importance [10]. Let's discuss this definition in parts.

Values are concepts about desirable end states or behaviors. Values are represented in the agent's mind as major guidelines that say what is good or bad, right or wrong and worth to achieve or not. They somehow define the character of the agent. Ambition, honesty and social recognition are some examples of values [14]. Each agent has its set of values (called values system). So, an agent may think that to have ambition is good while another agent may not. An agent may think that act with honesty is right while another agent may not consider it. An agent may think that is important to pursue social recognition while another agent may not care about it.

Values transcend specific situations. Different from other representations as goals and norms that are applied to specific situations and contexts, values work as general principles that are independent of time, place and objects [12]. For example, an agent that has honesty as an important value will act with honesty in every situation, because that is a priority to it.

Values guide the selection or evaluation of behavior and events. Similar to goals, we can say that values are reached by the execution of certain actions. Though, a value is not something that can be achievable [3]. How can we achieve ambition? How we can we say that we have enough social recognition? So, we say that a value is something that can be promoted or demoted¹ by certain actions and attitudes².

An agent will always prefer to execute an action or show a behavior that promotes its values. In the same way an agent will not intend to execute an action or show a behavior that demotes its values. In face of a choice, the agent will use its values system to evaluate the best option, i.e. which option will promote more and demote fewer values. Suppose that an agent who has the values honesty and social recognition finds a suitcase with a great amount of money. The agent will decide to return the suitcase to its owner, because not returning it would demote its honesty value. Then, the agent has two ways to return the suitcase: anonymously and non-anonymously. The agent will choose to return the suitcase non-anonymously because, in addition to promote the honesty value, it will also promote the social recognition value.

¹ A value can be promoted or demoted in many levels and the measurement of the intensity of the promotion or demotion is out of the scope of this paper because it is still an open question to Psychology [10].
² Some values can also be indirectly promoted/demoted by the promotion/demotion of other values [10], but the discussion of the effects of the promotion/demotion of values is out of the scope of this paper.
Values are ordered by relative importance. Although an agent can have a set of values, some values are more important to it than others, i.e. the priority of a value can be higher than another. Each value in an agent’s values system receives a relative importance that helps the agent reasoning to evaluate events and make decisions.

Consider that an agent who has the values ambition and honesty finds a suitcase with a great amount of money as in the previous example. If the agent returns the suitcase, then it will promote its honesty value and demote its ambition value. But, if it does not return the suitcase, then the ambition value will be promoted and the honesty value will be demoted. In cases like this, the importance of each value will be used to make the decision. If the importance of the ambition value is greater than the importance of the honesty value, the agent will not return the suitcase even knowing that this action will demote its honesty value.

When agents cohabit they share and discuss their values [12]. The common values of the majority of agents of one organization are part of the organizational culture [15] and are transmitted to the agents that inhabit such organization.

2.3 The Z Language

There are several languages and formal techniques available to specify properties of software systems [16]. In this work we choose to do the specification using the Z language [17]. Z is a model-oriented formal specification language based on set theory and first-order logic. By choosing the Z language we adopt a language that is extremely expressive, elegant and more accessible than many other formalisms because it is based on existing elementary components such as first order predicates and set theory. Moreover, Z is being used both in industry [18, 19] and academia, including MAS researches [20, 21]. It is also supported by a large array of books [22-24] and tools for type checking, testing and animating operations [25, 26].

The key syntactic element of Z is the schema, which allows specifications to be structured into manageable modular components. Z schemas consist of two parts: the upper declarative part, which declares variables and their types, and the lower predicate part, which relates and constrains those variables. If we allow \( d \) to stand for a set of declarations, \( p \) to be a set of predicates and \( S \) to be the name of a schema we have the basic notation for a schema as follows:

\[
\begin{array}{c}
S \\
d \quad \{\text{set of declarations}\} \\
p \quad \{\text{set of predicates}\}
\end{array}
\]

Declarations and predicates are composed by expressions using the elements of the Z language’s notation. A summary of the Z notation that is used in this paper is given in Fig. 1. More details about Z and its formal semantics can be found elsewhere [22].
Both norms and values are part of the organizational culture. On one hand the values support the agent's autonomy by representing its character and helping the agent to take its own decisions and execute actions based on what it thinks it's important and on its preferences. On the other hand the norms are used by the system to cope with the autonomy of the agents by regulating their behavior in order to promote the system order.

As stated before in Section 2.2, agents prefer to execute actions that promote their values and not execute actions that demote their values. Since norms may prohibit or oblige agents of execute actions, conflicts between the system's norms and the agent’s values may arise. For instance, if a norm prohibits the execution of an action that promotes a value or obligates the execution of an action that demotes a value, such norm is in conflict with the values of the agent.

The identification of the conflicts between the system's norms and the values is important to the agent when deciding, for instance: (i) whether or not to enter an organization once it realizes that the organization's norms are coherent or not with its values; and (ii) which actions it will execute and, as consequence, which norms it will violate or fulfill.

The identification of conflicts between norms and values at runtime is an important because new conflicts may arise at any time. First the norms change due to the dynamic of the MAS environments, and, second, the agent’s values may also change due to the living together and cultural influences [3, 10].
In Section 3.1 we discuss in which cases the norms and values are in conflicts. In the next sections we present the algorithm proposed to identify these conflict cases (Section 3.2) and an example of the application of this algorithm (Section 3.3). In our work we consider that the agent knows its values system, which actions promote and demote its values and which norms are applied to it.

### 3.1 Conflict Cases

Norms regulate the execution of actions by describing obligations, prohibitions and permissions to the agents. The actions to be executed by the agents can promote and demote some of their values. So, the actions are the central element linking norms to values, as shown in Fig. 2.

![Fig 2. Norm, Action and Value Relationships](image)

There may be norms that describe the actions that must be executed (obligations), the actions that cannot be executed (prohibitions), and the actions that can be executed (permissions). The agents will prefer to execute actions that promote their values and they will not intend to execute actions that demote their values. Considering this, we can point the two main conflict cases between norms and values, which are:

1. *The norm states an obligation to the agent to execute an action that demotes an important value to the agent;*
2. *The norm states a prohibition to the agent to execute an action that promotes an important value to the agent.*

In both cases the norm defines a behavior that is against the personal interests of the agent. Norms that state permissions to execute actions will not be in conflicts with the agent’s values, because even if the action regulated to the norm demotes an important value to the agent, it can simply choose not to execute it.

Despite the identification of conflicts between norms and values seems simple, it may be complicate to identify the conflicts when a norm and a value do not mention exactly the same action, but related actions. When a norm regulates an action and a value is promoted/demoted by another action, a simple conflict checker would conclude that the norm and the value are not in conflict since they do not mention the same action. However, if theses actions are related, the conflict may arise. In the next sections we detail the identification of conflict cases when one action is the specialization of another (items 3.1 and 3.2) and when one action is a part of another action (items 4.1 and 4.2).
3.1.1 Action Refinement

If the refinement relationship is defined between two actions, it means that there is an action called subaction that is more specific than another action called superaction. The states that should be achieved by executing the superaction are a subset of the states achieved by executing the subaction. If there is more than one subaction for a given superaction, it is necessary to execute only one subaction in order to achieve the goal of executing the superaction.

**Norm:** If the norm applied to the superaction is an obligation, it means that at least one subaction must be executed in order to fulfill the obligation. On the other hand, if the norm applied to the superaction is a prohibition, it means that if one subaction is executed the norm will be violated.

**Value:** The values promoted and demoted by a superaction are also promoted and demoted by all subactions. But each subaction can promote and demote new values. The relation between the action refinement relationship and its sets of promoted and demoted values are described in Fig. 3a.

So, if a norm regulates the execution of a superaction, there will be conflicts in the following cases:

3.1. **The norm states an obligation to the agent to execute a superaction and all its subactions demote an important value to the agent.** An obligation to execute a superaction means an obligation to execute one of its subactions. If all subactions demote an important value to the agent, then the agent will not be able to choose an action to fulfill the obligation without be in conflicts with its values system;

3.2. **The norm states a prohibition to the agent to execute a superaction and at least one of its subactions promotes an important value to the agent.** The prohibition to execute a superaction means a prohibition to execute each of its subactions. One subaction that promotes an important value to the agent is enough to establish a conflict between the norm and the values of the agent;

3.1.2 Action Composition

If the composition relationship is defined between two actions, it means that there is an action that is a part of another action called composed action. The states that should be achieved by executing the action that is part of the composition are a subset of the states achieved by executing the composed action. If there is more than one action in a given composition, it is necessary to execute all of them in order to achieve the goal of executing the composed action.

**Norm:** If the norm applied to the composed action is an obligation, it means that all actions of the composition must be executed in order to fulfill the obligation. On the
other hand, if the norm applied to the composed action is a prohibition, it means that only if all actions of the composition are executed together the norm will be violated.

**Fig 3.** The Promoted and Demoted Value Sets of Actions. – Let the set of promoted values of an action be \( PV \) and the set of demoted values of an action be \( DV \). The set of values promoted/demoted by a superaction is included in the set of values promoted/demoted by each of its subactions. And the set of values promoted/demoted by a composed action includes all the sets of values promoted/demoted by the actions of the composition.

**Value:** The values promoted and demoted by an action that is part of a composition are also promoted and demoted by the composed action. But the composed action can promote and demote new values. The relation between the action composition relationship and its sets of promoted and demoted values are described in Fig. 3b.

When the norm regulates the execution of a composed action, the conflict cases are:

4.1. *The norm states an obligation to the agent to execute a composed action and the composed action or at least one of the actions of the composition demotes an important value to the agent.* An obligation to execute a composed action means an obligation to execute all actions of the composition. If just one action of the composition demotes an important value to the agent, then there is a conflict. Also, there will be a conflict if the composed action itself demotes an important value to the agent;

4.2. *The norm states a prohibition to the agent to execute a composed action and the composed action or all actions of the composition promote an important value to the agent.* A prohibition to execute a composed action means a prohibition to execute all actions of the composition together. Then the conflict will arise only if all actions of the composition demote
an important value to the agent. Also, there will be a conflict if the composed action itself promotes an important value to the agent;

Just like the norms that regulate simple actions, there are no conflicts between values and permission norms that regulate superactions and composed actions.

3.2 An Algorithm to Identify the Conflicts

In this section we propose an algorithm to verify if a norm is in conflicts with the agent values by checking every case of conflict described in the previous section. The algorithm and its input elements (norm, value and action) are written in Z as functions and schema definitions.

Values are represented by the Value schema where identifier indicates the value identifier symbol from the set [ValueSym], i.e. the name of the value, and importance indicates the relative importance associated by the agent with that value. If importance = 0 then the value is not important to the agent and if importance > 0 then the agent consider the value important.

```
Value
identifier : ValueSym
importance : N
```

The actions are described by the Action schema where: identifier indicates the action identifier symbol from the set [ActionSym], i.e. the action’s name; subactions specifies the subactions set of the action, if there are any; compositeactions specifies the actions set of the composition, if the action is a composed action; promotes indicates the values set promoted by the action; and demotes indicates the values set demoted by the action. For consistence reasons, an action cannot be part of its own subactions or composite actions set. Also, an action cannot both promote and demote a given value.

```
Action
identifier : ActionSym
subactions : P Action
compositeactions : P Action
promotes : P Value
demotes : P Value

self \ cap \ subactions = Ø
self \ cap \ compositeactions = Ø
promotes \ cap \ demotes = Ø
```

3 The specification presented in this section can be considered as a preliminary extension of the specification of the BDI agent proposed in [20] to include values in the agent reasoning process.
Finally, norms are represented by the Norm schema where identifier indicates the norm identifier symbol from the set [NormSym]; deonticconcept indicates if the norm states an obligation, a prohibition or a permission as shown by DeonticConcept definition; and action indicates the action regulated by the norm. Although norms are usually composed by all the elements described in Section 2.1, we are only interested to address here the elements that affect the conflicts verification.

\[
\text{DeonticConcept} ::= \text{OBLIGATION} \mid \text{PROHIBITION} \mid \text{PERMISSION}
\]

Norm

- identifier : NormSym
- deonticconcept : DeonticConcept
- action : Action

The algorithm that we propose is divided in two parts or two functions: the checkNormValueMainConflicts and the checkNormValueAllConflicts. The first function verifies only the conflict cases 1 and 2 which just analyze the values promoted and demoted by a specific action regulated by the norm. The second function includes the first function and also covers the 3.1, 3.2, 4.1 and 4.2 conflict cases by analyzing all the values promoted and demoted by the subactions and composite actions of the structure of the action regulated by the norm.

The checkNormValueMainConflicts function maps a norm and its action to a boolean that indicates whether or not exists a conflict between the norm and the values promoted and demoted by the action of the norm. This function returns true if the deontic concept of the norm is obligation and at least one of the values demoted by the action of the norm has an importance greater than 0 (conflict case 1 – lines 1 to 3) or if the deontic concept of the norm is prohibition and at least one of the values promoted by the action of the norm has an importance greater than 0 (conflict case 2 – lines 4 and 5).

\[
\text{checkNormValueMainConflicts} : (\text{Norm} \times \text{Action}) \rightarrow \text{Boolean}
\]

1: \( \forall n : \text{Norm}; \ a : \text{Action} \rightarrow \text{checkNormValueMainConflicts}(n, a) \Leftarrow \)
2: \( (((n.\text{deonticconcept} = \text{OBLIGATION}) \land \)
3: \( \exists \text{demotedvalue} : a.\text{demotes} \mid \text{demotedvalue}.\text{importance} > 0)) \)
4: \( \lor (((n.\text{deonticconcept} = \text{PROHIBITION}) \land \)
5: \( \exists \text{promotedvalue} : a.\text{promotes} \mid \text{promotedvalue}.\text{importance} > 0)) \)

The checkNormValueAllConflicts is the main function. It is the function that must be called in order to check all the conflict cases between a norm and the values of the agent. The checkNormValueAllConflicts function maps a norm and its action to a boolean that indicates whether or not exists a conflict between the norm and the values promoted and demoted by the action of the norm and its subactions and composite actions.
First, the function verifies if the action inputted is really the action of the norm before proceeding the conflict identification (line 1). Then the function confirms if the action of the norm has subactions (line 2) in order to check recursively all the subactions and its values (line 2 to 8). The function returns true (i) if the deontic concept of the norm is obligation and a conflict was identified while checking all the subactions of the norm action (conflict case 3.1 – lines 3 and 4) or (ii) if the deontic concept of the norm is prohibition and a conflict was identified while checking at least one subaction of the norm action (conflict case 3.2 – lines 5 and 6).

After that, the function verifies if the action of the norm is a composed action (line 9) in order to check the action of the norm itself (line 10) and recursively verify all the actions of the composition and its values (line 9 to 19). The function returns true (i) if a conflict was identified while checking the action of the norm with the `checkNormValueAllConflicts` function, (ii) if the deontic concept of the norm is obligation and a conflict was identified while checking at least one action of the composition of the norm action (conflict case 4.1 – lines 12 to 14) or (iii) if the deontic concept of the norm is prohibition and a conflict was identified while checking all the actions of the composition of the norm action (conflict case 4.2 – lines 15 to 17).

\[\text{checkNormValueAllConflicts} : (\text{Norm} \times \text{Action}) \rightarrow \text{Boolean}\]

1: \(\forall n \in \text{Norm}; a \in \text{Action} | \text{checkActionsUnification}(a, n, \text{action})\) •
2: \(\text{if } a, \text{subactions} \neq \emptyset \text{ then}\)
3: \(\text{if } ((n, \text{deonticconcept}) = \text{OBLIGATION} \land\)
4: \((\forall \text{subaction} : a, \text{subactions} | \text{checkNormValueAllConflicts}(n, \text{subaction}))\)
5: \((\exists \text{subaction} : a, \text{subactions} | \text{checkNormValueAllConflicts}(n, \text{subaction})))\)
6: \(\text{then checkNormValueAllConflicts}(n, a) = \text{true}\)
7: \(\text{else checkNormValueAllConflicts}(n, a) = \text{false}\)
8: \(\text{else if } a, \text{composites} \neq \emptyset \text{ then}\)
9: \(\text{if checkNormValueMainConflicts}(n, a)\)
10: \(\text{then checkNormValueAllConflicts}(n, a) = \text{true}\)
11: \(\text{else if } ((n, \text{deonticconcept}) = \text{OBLIGATION} \land\)
12: \((\exists \text{compositeaction} : a, \text{composites} |\)
13: \(\text{checkNormValueAllConflicts}(n, \text{compositeaction}))\)
14: \(\lor (n, \text{deonticconcept}) = \text{PROHIBITION} \land\)
15: \(\exists \text{compositeaction} : a, \text{composites} |\)
16: \(\text{checkNormValueAllConflicts}(n, \text{compositeaction})))\)
17: \(\text{then checkNormValueAllConflicts}(n, a) = \text{true}\)
18: \(\text{else checkNormValueAllConflicts}(n, a) = \text{false}\)
19: \(\text{else checkNormValueAllConflicts}(n, a) = \text{checkNormValueMainConflicts}(n, a)\)

The function `checkActionsUnification` applies the unification between two actions as in [20 and 21] and was omitted here due to the lack of space. It can be seen in http://www.ic.uff.br/~kfigueiredo/values/normvalueconflictsidentification.pdf together with the complete specification presented in this work.
To conclude, if the action of the norm is not a superaction or a composition, the function calls the `checkNormValueMainConflicts` function and returns its result, applying the verification of the two main conflict cases to the single action and its values (conflict cases 1 and 2 - line 20).

The next section presents an example to illustrate our approach of conflicts identification between norms and values.

3.3 Running Example

Let’s consider an agent who has the values security and agility, both with importance = 1, i.e. both values are important to the agent. Suppose that exists a norm that obligates this agent to go from X to Y. Fig. 4 illustrates the paths the agent can use to go from point X to Y.

![Fig 4. Example Illustration](image)

The agent can go from X to Y via route A or via route B passing by Z. The route A is longer than route B, but the subroute B’ that connects Z to Y has some obstacles. On one hand, going through route A demotes the agility value since it is the longest path and going through route B promotes agility. On the other hand, going through route B’ demotes its security value because of the obstacles. The following structure describes the values, actions and norm of this scenario example according to the specification of the previous section:

```
security(1)
agility(1)
goFromXtoY(goFromXtoYViaA, goFromXtoYViaB, {}, {}, {})
goFromXtoYViaA({}, {}, {}, {agility})
goFromXtoYViaB({}, {goFromXtoZ, goFromZtoY}, {agility}, {agility})
goFromXtoZ({}, {}, {}, {}) 
goFromZtoY({}, {}, {}, {security})
```

```
norm1(OBLIGATION, goFromXtoY)
```
To check if norm1 is in conflict with the values of the agent, first the algorithm identifies that the action regulated by norm1, goFromXtoY, is a superaction. Since the norm is an obligation, the algorithm starts by analyzing all its subactions: goFromXtoYViaA and goFromXtoYViaB (checking conflict case 3.1).

The action goFromXtoYViaA is a simple action, so the algorithm verifies if this action demotes any value that is important to the agent (checking conflict case 1). In this case the action goFromXtoYViaA demotes the agility value, so the obligation of goFromXtoYViaA is in conflicts with the agent’s values (conflict case 1 identified).

Then, the algorithm identifies that the action goFromXtoYViaB is a composed action and starts to analyses each action of the composition: goFromXtoZ and goFromZtoY (checking conflict case 4.1). Both are simple actions and the algorithm verifies if any of them demotes a value important to the agent (checking conflict case 1). The action goFromXtoZ does not demote any value of the agent, but the action goFromZtoY demotes the security value due to the obstacles (conflict case 1 identified). So, the obligation of goFromXtoYViaB is in conflicts with the agent’s values because the agent would demote its security value in order to execute the composite action goFromZtoY (conflict case 4.1 identified).

As both obligations goFromXtoYViaA and goFromXtoYViaB are in conflicts with the agent’s values, the obligation of norm1 to execute goFromXtoY is also in conflicts with the agent’s values because the agent is not able to fulfill the obligation without demote any of its values (conflict case 3.1 identified). The presented example covered the algorithm execution and three conflict cases between norms and values pointed out in this paper.

4 Conclusion and Future Work

In this paper we presented the conflict cases that can arise between the system’s norms and the values of the agents. We have also provided an algorithm that can identify these conflicts and support the different relationships between norms, values and action types. As far as we know, this is the first MAS work that investigates the identification of conflicts between norms and values.

The identification and resolution of the conflicts between the system's norms and the values are important to the agents in their decision making process. The agents can choose the actions to execute, the norms to fulfill or violate and the groups to join.

This research is the beginning of a process. The next steps of this work consist of develop ways to help the agents to solve the norms-values conflicts (for instance, by using the importance value to support the decisions as in the last example of Section 2.2) and make a full extension of the BDI agent architecture proposed in [20, 21] in order to include values and its influences in the agent’s reasoning process.
References