Noname manuscript No.
(will be inserted by the editor)

A Coherence Maximisation Process For Solving Normative Inconsistencies

Natalia Criado · Elizabeth Black · Michael Luck

the date of receipt and acceptance should be inserted later

Abstract Norms can be used in multi-agent systems for defining patterns of behaviour in terms of permissions, prohibitions and obligations that are addressed to agents playing a specific role. Agents may play different roles during their execution and they may even play different roles simultaneously. As a consequence, agents may be affected by inconsistent norms; e.g., an agent may be simultaneously obliged and forbidden to reach a given state of affairs. Dealing with this type of inconsistency is one of the main challenges of normative reasoning. Existing approaches tackle this problem by using a static and predefined order that determines which norm should prevail in the case where two norms are inconsistent. One main drawback of these proposals is that they allow only pairwise comparison of norms; it is not clear how agents may use the predefined order to select a subset of norms to abide by from a set of norms containing multiple inconsistencies. Furthermore, in dynamic and non-deterministic environments it can be difficult or even impossible to specify an order that resolves inconsistencies satisfactorily in all potential situations. In response to these two problems, we propose a mechanism with which an agent can dynamically compute a preference order over subsets of its competing norms by considering the coherence of its cognitive and normative elements. Our approach allows flexible resolution of normative inconsistencies, tailored to the current circumstances of the agent. Moreover, our solution can be used to determine norm prevalence among a set of norms containing multiple inconsistencies.

Keywords norm inconsistencies · coherence · BDI agents

Natalia Criado
Liverpool John Moores University, UK
E-mail: n.criado@ljmu.ac.uk

Elizabeth Black
King's College London, UK
E-mail: elizabeth.black@kcl.ac.uk

Michael Luck
King's College London, UK
E-mail: michael.luck@kcl.ac.uk

1 Introduction

Norms are used in multi-agent systems to define control and coordination mechanisms intended to influence the behaviour of autonomous and heterogeneous agents [30]. In this paper, norms specify the expected behaviour of roles in terms of obligations, permissions and prohibitions. In particular, we regard normative specifications as being conditional expressions that specify under which circumstances instances of norms must be created and deleted (i.e., when instances become active and when they expire) [25,30,35,11]. The circumstances as well as the roles played by agents may change at execution time. Therefore, the instances that apply to agents are a priori unknown. Moreover, there may be inconsistencies[1] among these instances; e.g., an agent may be simultaneously obliged and forbidden to bring about a given state of affairs. Since inconsistency only arises at execution time, agents must be endowed with mechanisms for resolving inconsistency when it arises. Dealing with this type of inconsistency is a key challenge of normative reasoning.

Existing work concerning normative agents typically proposes the use of a static order based on norm salience (i.e., importance of norms) to determine which instance prevails in the case of inconsistency (i.e., the instance created out of the most salient norm prevails) [7, 12,4,41]. However, a significant drawback of such work is the fact that the order is specified off-line and is hard-wired into agents. Thus, the work assumes that it is possible to specify an order that appropriately resolves any inconsistency that may arise at execution time. This assumption is too strong for dynamic and non-deterministic environments in which the performance of the system may be unpredictable. In these circumstances it can be difficult, or even impossible, to specify an order that ensures that inconsistencies are resolved satisfactorily in any situation. This may be even more complicated in systems in which norms can be changed on-line. Thus, agents may be in a situation in which they must resolve an inconsistency among instances that have been created out of norms defined at run-time. In addition, agents should resolve normative inconsistencies when there are not only several inconsistency relationships among a set of instances but also cognitive elements that can support or oppose these instances. In these kinds of situations, it is not obvious how to use a static, predefined, order over norms to determine which of the instances should prevail.

In response, in this paper we propose to endow Normative Graded BDI agents [11] with a mechanism to dynamically resolve inconsistencies among several instances based on *coherence theory* principles [39]. Coherence is a cognitive theory whose main purpose is the study of associations; i.e., how representational elements (e.g., cognitive elements and normative elements) influence each other by imposing a positive or negative constraint over the rest of the representational elements. Coherence dynamically computes a preference order over the power set of instances by considering the constraints among a set of representational elements. Therefore, our coherence-based solution can adapt to different cognitive states depending on the information that an agent has at its disposal. Moreover, it can be used to select a subset of coherent instances from a set of instances containing multiple inconsistencies. We analyse the performance of our solution for resolving inconsistencies across different examples and experiments. With this information, we determine which kinds of applications may potentially benefit from using our solution.

In this paper we use a running example to: (i) motivate the need for our coherence maximisation process to solve normative inconsistencies; and (ii) illustrate how an agent carries out the different steps of the coherence maximisation process to resolve inconsistencies in a particular situation. Let us consider a case study of the management of websites in a

[1] In this paper we will use the terms inconsistency and normative inconsistency as synonyms.

university. Specifically, let us assume the existence of a *webManager* agent, in charge of dynamically distributing websites into two servers that are maintained by the university. Note that the *webManager* agent is not the addressee of the norms, which is instead the person that created the websites being maintained by the *webManager*. However, the *webManager* agent must reason about the norms that are addressed to the website creator. Thus, the *webManager* agent acts as a proxy for the website creator[2]. One of these servers, identified by $slow$, is dedicated to maintaining personal websites of students, academics, etc. The other server, identified by $fast$, is a faster server that is dedicated to maintaining websites that are of high importance to the university; e.g., conference websites, e-learning sites, etc. There are four norms that specify which server must or must not be used in each particular situation: (i) there is a norm that obliges the *webManager* agent to use the server that is not under maintenance at a given moment; (ii) there is a general prohibition against using $fast$ since it must be devoted to important pages; (iii) when a webpage belongs to an academic and $slow$ is overloaded, then $fast$ can be used; and (iv) it is forbidden to store webpages on any server when the security of the whole system is compromised. In what follows, we illustrate that it is possible for instances of these norms to be active simultaneously, so that the *webManager* agent must make a decision about which instance prevails.

The paper is structured as follows: Section 2 provides the necessary definitions, while Section 3 provides an overview of coherence theory; in Section 4 we instantiate coherence theory for the problem of resolving normative inconsistencies; Section 5 assesses the performance of our proposal for resolving inconsistencies between two instances in different situations; Section 6 presents an experimental analysis of performance for inconsistencies among larger sets of instances; and, finally, a discussion of related work and conclusions are contained in Sections 7 and 8, respectively.

2 Preliminary Definitions

2.1 Basic Definitions

We make use of a first-order predicate language $\mathcal{L}$ whose alphabet includes: the logical connectives $\{\wedge, \vee, \neg\}$; equality and inequality symbols; the true ($\top$) and false propositions ($\bot$); an infinite set of variables; and predicate, constant and function symbols. Variables are implicitly universally quantified[3]. In this paper, variables are written as any sequence of alphanumeric characters beginning with a capital letter. Predicate, constant and function symbols are written as any sequence of alphanumeric characters beginning with a lower case letter. Let us assume the standard definition for well-formed formulas (*wffs*). We make use of the standard notion of substitution of variables in a *wff*; i.e., σ is a finite and possibly empty set of pairs Y/y where Y is a variable and y is a term [20]. $\mathscr{R}$ and $\mathscr{A}$ are the sets containing all role and agent identifiers, respectively. In particular, $self \in \mathscr{A}$ is an agent identifier representing the agent that is performing the reasoning process. For the purpose of this paper it is necessary to know that the relationship between agents and roles is formally represented by a binary predicate ($play$). Specifically, the expression $play(a, r)$ describes the fact that the agent identified by $a \in \mathscr{A}$ enacts the role identified by $r \in \mathscr{R}$.

[2] The existence of normative agents that mediate between external agents or humans and multi-agent systems is not new. For example, in Electronic Institutions [19] there are *governor* agents that guarantee that external agents comply with the norms of the institution.

[3] In this paper, we regard norms as conditional expressions that specify under which general circumstances they must be instantiated.

2.2 Normative Definitions

Norms help to define control, coordination and cooperation mechanisms that attempt to: (i) promote behaviours that are satisfactory to the organisation, i.e., actions that contribute to the achievement of global goals; and (ii) avoid harmful actions, i.e., actions that may lead to an undesirable state. Norms have been studied from different perspectives such as philosophy [43], sociology [36], law [1], etc., and multi-agent systems research has given different meanings to the norm concept. For example, it has been employed as a synonym of obligation and authorization [14], social law [33], social commitment [38] and other kinds of rules imposed by societies or authorities. The purpose of this paper is not to propose, compare or improve existing normative definitions, but to make use of these definitions in proposing a coherence-based mechanism to allow agents to resolve normative inconsistencies. The aim of this section is to provide the basic normative notions used in the paper.

In particular, in this paper we consider *norms* as formal statements that define patterns of behaviours by means of *deontic modalities* (i.e., *obligations*, *permissions* and *prohibitions*). Specifically, our proposal is based on the notion of a norm as a general rule of behaviour that defines under which circumstances a pattern of behaviour must be instantiated. This notion of norm has been widely used in the existing literature (e.g., [25], [30], [35] and [11]).

Definition 1 (Norm) A norm is defined as a tuple $n = \langle \Delta, C, T, A, E \rangle$, where:

- $\Delta \in \{\mathcal{O}, \mathcal{F}, \mathcal{P}\}$ is the deontic modality of the norm, determining if the norm is an obligation ($\mathcal{O}$), prohibition ($\mathcal{F}$) or permission ($\mathcal{P}$);
- C is a wff of $\mathcal{L}$ that represents the norm condition on the affected agents, i.e., it denotes the state of affairs that the target of the norm must do/bring about (in the case of obligations), refrain from generating (in the case of prohibitions), or is permitted to do/bring about (in the case of permissions);
- $T \in \mathscr{R}$ is the target of the norm; i.e., the role to which the norm is addressed;
- A is a wff of $\mathcal{L}$ that describes the activation condition;
- E is a wff of $\mathcal{L}$ that describes the expiration condition.

For example, the norm that obliges use of the server that is not being maintained is represented as follows:

$$\langle \mathcal{O}, use(S1), universityMember, maintenance(S2) \wedge S1 \neq S2, \neg maintenance(S2) \rangle$$

Similarly, the norm that forbids the use of $fast$ is represented as follows:

$$\langle \mathcal{F}, use(fast), universityMember, \top, \bot \rangle$$

The norm that permits academics to use $fast$ when $slow$ is overloaded defined as follows:

$$\langle \mathcal{P}, use(fast), academicStaff, highTraffic(slow), lowTraffic(slow) \rangle$$

Finally, the norms that forbids the use of any server in case of a security problem is formalised as follows:

$$\langle \mathcal{F}, use(fast) \wedge use(slow), universityMember, securityThreat, \neg securityThreat \rangle$$

Once the activation condition of a norm holds several instances, according to the groundings of the activation condition (i.e., the sets of bindings of variables that occur on the activation condition), are created. Thus, an instance is an unconditional expression that binds agents playing a given role to an obligation, prohibition or permission. Instances remain active until their expiration condition holds.

Definition 2 (Instance) Given a norm $n = \langle \Delta, C, T, A, E \rangle$ and a knowledge base Γ of $\mathcal{L}$, an instance of n under the substitution σ is defined as the tuple $i = \langle \Delta, \bar{C}, T, \bar{A}, \bar{E} \rangle$ where:

- σ is a substitution of variables in A such that $\Gamma \vdash \sigma \cdot A$ and $\sigma \cdot C$, $\sigma \cdot A, \sigma \cdot E$ are grounded;
- $\bar{A} = \sigma \cdot A$, $\bar{E} = \sigma \cdot E$, $\bar{C} = \sigma \cdot C$.

Both the activation and the expiration conditions of norms might be undefined. When the activation condition is undefined (i.e., when A is $\top$), the norm is instantiated by default and an instance of this norm is always active. When the expiration condition is undefined (i.e., when E is $\bot$), there is no state of affairs that causes instances created out of this norm to expire. For a complete description of the dynamics and operational semantics of norms and instances within a multi-agent system see [9].

For example, if we assume that $slow$ is maintained (i.e., $\Gamma \vdash maintenance(slow)$), then the previous obligation norm is instantiated as follows.

$$\langle \mathcal{O}, use(fast), universityMember, maintenance(slow), \neg maintenance(slow) \rangle$$

To ensure that all instances have no free variables, we define the notion of well-formed norm as follows.

Definition 3 (Well-Formed Norm) A norm $n = \langle \Delta, C, T, A, E \rangle$ is a well-formed norm iff $V_A \supseteq V_E \cup V_C$; where V_X is the set of variables occurring in wff X.

2.2.1 Instance Inconsistency

Intuitively, we consider that two instances are inconsistent when they prescribe patterns of behaviours that are contradictory (i.e., logically incompatible). In particular, two obligation instances are inconsistent if fulfilling them both would lead to a contradiction (i.e., if the two instances have norm conditions that are logically incompatible); the same is true of two prohibition instances. An inconsistency between a permission instance and an obligation/prohibition instance exists if achieving the norm condition of the permission and fulfilling the obligation/prohibition would lead to a contradiction. Similarly, an obligation instance and a prohibition instance are inconsistent if fulfilling them both would lead to a contradiction. Given that permissions do not prescribe that agents must or must not bring about states of affairs, a situation in which an agent is permitted to bring about contradictory states is not defined as an inconsistency. More formally, we define inconsistency of instances as follows.

Definition 4 (Inconsistency of instances) Two instances $\langle \Delta^1, \bar{C}^1, T^1, \bar{A}^1, \bar{E}^1 \rangle$ and $\langle \Delta^2, \bar{C}^2, T^2, \bar{A}^2, \bar{E}^2 \rangle$ are inconsistent iff one of the following conditions holds:

- $\Delta^1 = \mathcal{O}$, $\Delta^2 = \mathcal{O}$ and $\{\bar{C}^1, \bar{C}^2\} \vdash \bot$;
- $\Delta^1 = \mathcal{F}$, $\Delta^2 = \mathcal{F}$ and $\{\neg\bar{C}^1, \neg\bar{C}^2\} \vdash \bot$;
- $\Delta^1 = \mathcal{O}$, $\Delta^2 = \mathcal{P}$ and $\{\bar{C}^1, \bar{C}^2\} \vdash \bot$;
- $\Delta^1 = \mathcal{P}$, $\Delta^2 = \mathcal{O}$ and $\{\bar{C}^1, \bar{C}^2\} \vdash \bot$;
- $\Delta^1 = \mathcal{P}$, $\Delta^2 = \mathcal{F}$ and $\{\bar{C}^1, \neg\bar{C}^2\} \vdash \bot$.
- $\Delta^1 = \mathcal{F}$, $\Delta^2 = \mathcal{P}$ and $\{\neg\bar{C}^1, \bar{C}^2\} \vdash \bot$.
- $\Delta^1 = \mathcal{O}$, $\Delta^2 = \mathcal{F}$ and $\{\bar{C}^1, \neg\bar{C}^2\} \vdash \bot$;
- $\Delta^1 = \mathcal{F}$, $\Delta^2 = \mathcal{O}$ and $\{\neg\bar{C}^1, \bar{C}^2\} \vdash \bot$;

As explained above, the cases in the definition cover all possible inconsistency relationships between two instances, and that no other inconsistency relationships are possible.

In our example, there is a norm that forbids the use of $fast$. Obviously, an instance of this norm is inconsistent with the instance that obliges the *webManager* agent to use $fast$ due to the fact that $slow$ is being maintained.

In this paper, we use a "closed legal system", which is a normative system where everything is considered as permitted by default. Permissions specify exceptions to the application of more general obligations and prohibitions. Thus, a permission creates an inconsistency with a more general instance by identifying specific situations in which the more general instance does not apply. In our example, suppose that the *webManager* agent is dealing with a website belonging to an academic and that $slow$ is overloaded with visits. In this case the norm that permits use of $fast$ is instantiated. According to our definition, this instance is inconsistent with the instance that forbids the use of $fast$.

2.2.2 Instance Satisfaction

Intuitively, we consider that one instance satisfies another when they are in accordance (i.e., when one instance prescribes a pattern of behaviour that entails the pattern of behaviour prescribed by other instance). In particular, two obligation, prohibition or permission instances satisfy each other when they are in accordance (i.e., if the two instances have norm conditions that are related by logical entailment). A permission instance and an obligation/prohibition instance satisfy each other when fulfilling the obligation/prohibition entails achieving the norm condition of the permission, and vice versa. Similarly, an obligation instance and prohibition instance satisfy each other when fulfilling one of the instances entails the fulfilment of the other instance. And there are no more satisfaction relationships between two instances. More formally, we define satisfaction of instances as follows.

Definition 5 (Satisfaction of one instance by another) The instance $\langle \Delta^1, \bar{C}^1, T^1, \bar{A}^1, \bar{E}^1 \rangle$ satisfies the instance $\langle \Delta^2, \bar{C}^2, T^2, \bar{A}^2, \bar{E}^2 \rangle$ iff one of the following conditions holds:

- $\Delta^1 = \mathcal{F}$, $\Delta^2 = \mathcal{F}$ and $\neg\bar{C}^1 \vdash \neg\bar{C}^2$;
- $\Delta^1 = \mathcal{O}$, $\Delta^2 = \mathcal{O}$ and $\bar{C}^1 \vdash \bar{C}^2$;
- $\Delta^1 = \mathcal{P}$, $\Delta^2 = \mathcal{P}$ and $\bar{C}^1 \vdash \bar{C}^2$;
- $\Delta^1 = \mathcal{O}$, $\Delta^2 = \mathcal{P}$ and $\bar{C}^1 \vdash \bar{C}^2$;
- $\Delta^1 = \mathcal{P}$, $\Delta^2 = \mathcal{O}$ and $\bar{C}^1 \vdash \bar{C}^2$;
- $\Delta^1 = \mathcal{F}$, $\Delta^2 = \mathcal{P}$ and $\neg\bar{C}^1 \vdash \bar{C}^2$;
- $\Delta^1 = \mathcal{P}$, $\Delta^2 = \mathcal{F}$ and $\bar{C}^1 \vdash \neg\bar{C}^2$.
- $\Delta^1 = \mathcal{O}$, $\Delta^2 = \mathcal{F}$ and $\bar{C}^1 \vdash \neg\bar{C}^2$;
- $\Delta^1 = \mathcal{F}$, $\Delta^2 = \mathcal{O}$ and $\neg\bar{C}^1 \vdash \bar{C}^2$;

As explained above, the cases in the definition cover all possible satisfaction relationships between two instances, and no other satisfaction relationships are possible.

In our example, the instance that permits the use of $fast$ for academic webpages when $slow$ is overloaded satisfies the instance that obliges the use of $fast$ (due to the maintenance of $slow$). Furthermore, let us suppose that an attacker intrusion has been detected. In this situation the security protection norm is instantiated and, as a consequence, the use of either $fast$ or $slow$ is forbidden. This instance satisfies the instance that forbids the use of $fast$.

In this paper we propose the use of coherence theory to deal with situations in which agents are affected not only by inconsistent instances but also by instances that satisfy another.

2.3 Normative Agent Definition

We assume a practical reasoning agent [3] whose actions are controlled by norms and directed towards its goals. Specifically, we focus on how an agent resolves the inconsistencies among the instances that affect it. To make such decisions, we propose that the agent considers: the salience of norms (which has been defined in [5] as the degree of activity and importance of a norm within a social group and a given context) that have given rise to the instances, the ease of compliance of these instances (i.e., how difficult it is for an agent to comply with these instances), the agent environment (i.e., the certainty of the beliefs about the agent's current circumstances) and the impact of instances (i.e., the importance of the desires hindered or favoured by the instances). In order to capture these different parameters, and to allow our agents to deal with dynamic and non-deterministic environments, we use the Normative Graded BDI architecture (known as n-BDI) [11], which extends the Graded BDI architecture proposed by Casali et al. in [6] with an explicit representation of norms and instances.

Definition 6 (n-BDI Agent) An n-BDI agent is defined as a tuple $\langle B, D, I, N, N_I \rangle$, where:

- B, D, I are the sets of graded beliefs, desires and intentions of the agent. These sets are composed of $M(\gamma, \rho)$ expressions, where: $M \in \{belief, desire, intention\}$ is a graded modality used for representing graded beliefs, desires or intentions, respectively; γ is a grounded formula of $\mathcal{L}$; and $\rho \in [0, 1]$ represents the degree associated with this mental formula. ρ represents a certainty degree in case of belief, a desirability degree in case of desires[4], and an intentionality degree in case of intentions[5].
- N is a set formed by $norm(n, \rho_s)$ expressions, where n is a norm, and $\rho_s \in [0, 1]$ is a real value that assigns a salience to this norm. This salience represents the importance of the norm.
- N_I is a set composed of $instance(i, \rho_c)$ expressions, where i is an instance, and $\rho_c \in [0, 1]$ is a real value that specifies the ease of compliance of the instance. This ease of compliance represents how difficult it is for the agent to comply with the instance.

Thus, the sets B, D, and I contain the cognitive elements, whereas the sets N and N_I contain the normative elements.

Our aim is not to provide a complete description of the reasoning process performed by n-BDI agents, the complete details of which can be consulted in [11][6]. For the purpose of this paper, it is only necessary to know that the reasoning process of n-BDI agents is mainly performed by deductive rules that connect cognitive and normative elements. In particular, the information flows from perception to action according to three main steps. Firstly, the agent *perceives* the environment and updates its beliefs, norms and instances accordingly. In particular, the agent revises the norms that are in force in its environment (i.e., the norms that have been established and not abolished). It then creates new instances out of the norms for which the activation conditions have become true and removes those instances that have expired. Secondly, in the *deliberation* step, the desire set is revised (e.g., new desires may

[4] As defined in [6], the desirability of a formula γ represents to what extent an agent wants to achieve a situation in which γ holds.

[5] According to [6], intentions are not considered as a basic attitude. Thus, the intentions of n-BDI agents are generated on-line from the agents' beliefs and desires. The intentionality degree of a formula γ is the consequence of finding a best feasible plan that permits a state of the world where γ holds to be achieved.

[6] See [10] for the pseudocode of the algorithm executed by n-BDI agents.

be created out of the user preferences). Also as part of the deliberation step, the agent considers the set of instances and makes a decision about which to comply with. This decision about compliance entails dealing with any inconsistencies among the active instances. The mechanism proposed in this paper allows these n-BDI agents to dynamically resolve these inconsistencies. The instances that the agent decides to comply with are translated into desires. Finally, in the *decision making* step, desires and beliefs are considered for deriving intentions.

Regarding the procedures by which norms are added and removed from N, for the purpose of this paper it is only necessary to assume that an n-BDI agent is endowed with procedures for maintaining a norm base that contains current norms; i.e., the norms that are *in force* at a given moment. For example, norms and their salience can be specified *off-line* by the designer of the *normative system* [1], who determines the importance of norms in the norm hierarchy. Norms and their salience can also be determined *on-line* by analysing perceptions that are relevant to norm recognition. These perceptions can be *explicit normative perceptions*, which correspond to those messages exchanged by agents in which norms and their salience are explicitly communicated (e.g., in [11] the authors propose methods for inferring norms and their salience by combining the information sent by multiple experts); or *implicit normative perceptions*, which correspond to the observation of actions performed by agents (e.g., where the inference of norms and their salience is based on imitation approaches [17], or on an analysis of normative signals such as punishments, sanctions and rewards [42]).

With regard to the set of instances N_I, we assume that n-BDI agents are endowed with procedures for adding new instances and removing expired instances from N_I. In particular, an instance is created whenever there is some substitution that causes the grounded activation condition (under that substitution) of the norm to be entailed from the agent's beliefs and a belief that represents that the agent enacts the target role of the norm (in the following we refer to this type of belief as *addressing beliefs*). When an instance is created, the agent estimates the ease of compliance by considering the difficulty of complying with the instance. In this paper, we will assume that n-BDI agents know the effects of actions; i.e., they have beliefs such as $\mathit{belief}(\mathit{effect}(\alpha, \gamma), \rho)$ that represents that the agent knows that action α causes γ with a certainty degree ρ. Thus, the ease of compliance with a particular instance can be calculated at run-time simply by considering the certainty with which available actions fulfil the instance. For example, an agent that is affected by an obligation to store web pages on a server that it knows to be unavailable (i.e., there is no action to save files to the server) may determine that the ease of compliance of this instance is 0 and, as a consequence, it has no chance of complying with the instance. In [8] the authors have proposed more elaborate methods for calculating the ease of compliance according to the impact of instances on the agent goals and emotions. Finally, instances are removed from N_I when the expiration condition is entailed from the agent's beliefs. Note that instances are not removed once the agent no longer believes it is playing the target role, which models the fact that instances created under some role must be fulfilled even if the agent stops enacting this role. For example, if a seller has contracted an obligation to deliver an item to a buyer who has paid for this item, then the seller must deliver the item even if he stops being a seller. However, norms that expire once agents stop enacting the target role can also be represented by redefining the expiration condition as a disjunction between the expiration condition and the addressing belief.

Table 1 shows an example of the formulas present in knowledge base of the *webManager* agent at some point of its execution. For simplicity, we ignore the contents of the cognitive sets and focus on the normative sets. The behaviour of the *webManager* agent is regulated

by the four norms that are contained in the set N, labelled as n_1, n_2, n_3 and n_4 in the N row of Table 1. Let us assume that the four norms have been instantiated with the maximum ease of compliance (indicating that there are no barriers to complying with any of the instances individually), creating the instances i_1, i_2, i_3 and i_4 in row N_I of Table 1.

Sets	**Content**
B	....
D	...
I	...
N	$norm(n_1, 1), norm(n_2, 0.6), norm(n_3, 0.3), norm(n_4, 0.8)$
N_I	$instance(i_1, 1), instance(i_2, 1), instance(i_3, 1), instance(i_4, 1)$

where $n_1 = \langle \mathcal{O}, use(S1), universityMember, maintenance(S2) \wedge S1 \neq S2, \neg maintenance(S2) \rangle$
$n_2 = \langle \mathcal{F}, use(fast), universityMember, \top, \bot \rangle$
$n_3 = \langle \mathcal{P}, use(fast), academicStaff, highTraffic(slow), lowTraffic(slow) \rangle$
$n_4 = \langle \mathcal{F}, use(fast) \vee use(slow), universityMember, securityThreat, \neg securityThreat \rangle$
$i_1 = \langle \mathcal{O}, use(fast), universityMember, maintenance(slow), \neg maintenance(slow) \rangle$
$i_2 = \langle \mathcal{F}, use(fast), universityMember, \top, \bot \rangle$
$i_3 = \langle \mathcal{P}, use(fast), academicStaff, highTraffic(slow), lowTraffic(slow) \rangle$
$i_4 = \langle \mathcal{F}, use(fast) \vee use(slow), universityMember, securityThreat, \neg securityThreat \rangle$

Table 1: Knowledge base of the *webManager* agent

Among these instances there are satisfaction and inconsistency relationships. We also consider there to be a satisfaction relationship between an instance and the norm from which it was created. Figure 1 shows these relationships among the instances and norms. Instances are represented by ellipses labelled with the instance itself and its ease of compliance. These instances have been created out of four norms that are also represented by ellipses labelled with the norm and the salience of the norm. Satisfaction (vs. inconsistency) relationships among these normative elements are represented by continuous (vs. dashed) lines. As depicted by the figure, the *webManager* agent faces a situation in which it is affected by several consistent and inconsistent instances, which have been created out of norms with different salience values. This is an example of a situation in which it is not straightforward to decide which instance or instances should prevail[7]. Furthermore, in this example we have not considered the relationships between instances and cognitive elements (e.g., desires that may either support or oppose the fulfilment of instances), which would further complicate the decision were they to be taken into account.

In this paper we address complex situations like that illustrated by our example, in which agents should resolve normative inconsistencies when there are not only several inconsistency and satisfaction relationships among a set of instances but also cognitive elements that can support or oppose these instances. In these kinds of situations, it is not obvious how to use a static, predefined salience order over norms to determine which of the instances should prevail. To address this problem, we propose a coherence maximisation process for allowing n-BDI agents to resolve normative inconsistencies in a dynamic way that adapts to current circumstances depending on the cognitive elements available.

[7] In this case the static order will determine that just the instance created out of the most salient norm should prevail. In Section 5 we demonstrate that approaches that only rely on salience to solve normative inconsistencies can lead to undesired results, even if only two instances are considered.

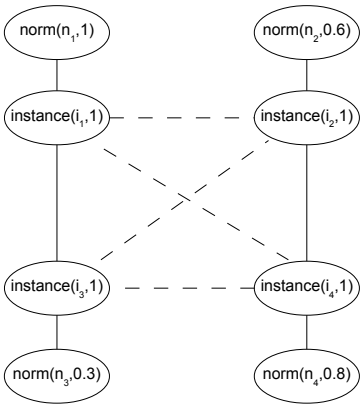


Fig. 1: Instances and norms affecting the *webManager* agent. Ellipses represent the normative elements. Coherence relationships among these elements are represented by continuous lines, whereas dashed lines represent incoherence relationships.

3 Coherence Theory

In [39] Thagard claims that coherence is a cognitive theory whose main purpose is the study of associations; i.e., how representational elements (cognitive and normative elements in our case) influence each other by imposing a positive or negative constraint over the rest of the elements. In [40] coherence is explained in terms of maximal satisfaction of multiple constraints. In particular, a coherence problem is formalised as follows. Let $\mathcal{V}$ be a finite set of elements $\{v_i\}$ which may be propositions or other representations, and $\mathcal{C}$ a set of constraints on $\mathcal{V}$ understood as a set $\{(v_i, v_j)\}$ of pairs of elements of V representing coherence and incoherence relationships between elements. $\mathcal{C}$ divides into $\mathcal{C}+$, the positive constraints (i.e., the coherence relationships) on $\mathcal{V}$, and $\mathcal{C}-$, the negative constraints (i.e., the incoherence relationships) on $\mathcal{V}$. There is a function ζ that associates with each constraint a number, which is the weight (strength) of the coherence or incoherence relationship. The problem is to partition $\mathcal{V}$ into two sets, accepted $\mathcal{A}$ and rejected $\mathcal{R}$, in a way that maximises compliance with the following two *coherence conditions*:

- if (v_i, v_j) is in $\mathcal{C}+$ then v_i is in $\mathcal{A}$ if and only if v_j is in $\mathcal{A}$;
- if (v_i, v_j) is in $\mathcal{C}-$ then v_i is in $\mathcal{A}$ if and only if v_j is in $\mathcal{R}$.

The coherence problem is to partition $\mathcal{V}$ into $\mathcal{A}$ and $\mathcal{R}$ in a way that maximises the coherence of the partition, that is, the sum of the weights of satisfied constraints.

In [40], the authors prove that the problem of calculating coherence is NP-complete and they provide several algorithms for maximising coherence.

- *Exhaustive Search.* The simplest way of maximizing coherence is to consider all the different ways of accepting and rejecting elements and calculate the coherence of each. The problem of this approach is that for a set $\mathcal{V}$ with $|\mathcal{V}|$ elements, there are $2^{|\mathcal{V}|}$ possible acceptance sets, which makes this approach intractable for reasonably sized problems.
- *Incremental Algorithm.* This is a simple serial algorithm in which an arbitrary ordering is applied to the elements. For each element in the ordering, if adding it to $\mathcal{A}$ increases the total weight of satisfied constraints more than adding it to $\mathcal{R}$, then it is added to $\mathcal{A}$;

otherwise it is added to $\mathcal{R}$. Obviously, this algorithm does not guarantee that the optimal solution is found, but it can be used to model bounded rationality [37].
- *Greedy Algorithm.* This algorithm starts with a randomly generated solution and then improves it by flipping elements from the accepted set to the rejected set or vice versa. This algorithm has been demonstrated to have good performance in several coherence problems [40].
- *Connectionist Algorithm.* This algorithm uses a neural network to assess coherence. While there are no mathematical guarantees on the quality of the solutions, empirical results yield excellent results for problems of a variety of sizes [40] (e.g., the connectionist algorithm has been used to solve problems with more than 400 elements and more than 10000 coherence links).
- *Semidefinite Programming Algorithm.* This algorithm is guaranteed to satisfy a high proportion of the maximum satisfiable constraints. In particular, the weight of the constraints satisfied by a solution will be at least 0.878 times the weight of the constraints satisfied by the optimal solution.

In his work, Thagard solved different kinds of coherence problems following this constraint maximisation approach. Given that in n-BDI agents the relationships among cognitive and normative elements are defined in terms of deductive rules, we focus on deductive coherence, which is concerned with coherence among logical propositions that belong to a deductive system. Next, the deductive coherence principles are explained in detail.

3.1 Deductive Coherence

According to Thagard's definition, *deductive coherence* [39] is a coherence problem whose elements are related by *deductive coherence* (ζ) yielded by propositional logical deduction. There are five principles that establish relations of deductive coherence and that allow the global coherence of a deductive system to be assessed. Given P, Q and $P_1, ..., P_n$ as propositions of a deductive system S, the principles of deductive coherence [39] are as follows.

1. *Symmetry.* Deductive coherence is a symmetric relation.
2. *Deduction.* If $P_1, ..., P_n$ deduce Q, then:
   (a) Any proposition coheres with propositions that are deductible from it. Thus, for each P_i in $\{P_1, ..., P_n\}$, P_i and Q cohere.
   (b) Propositions that together are used for deducing some other proposition cohere with each other. For each P_i and P_j in $\{P_1, ..., P_n\}$ P_i and P_j cohere.
   (c) The more hypotheses it takes to deduce something, the less the degree of coherence. Thus, in (a) and (b) the degrees of coherence are inversely proportional to n.
3. *Intuitive Priority.* Propositions that are intuitively obvious have a degree of acceptability on their own.
4. *Contradiction.* Contradictory propositions are incoherent with each other.
5. *Acceptability.* The acceptability of a proposition in a system of propositions depends on its coherence with them.

Once the general notion of deductive coherence has been provided, it is necessary to adapt this general theory in order to address the particular problem of resolving inconsistencies among normative instances.

4 Coherence for Solving Inconsistencies

The formalisation described in this section has been inspired by the work of Joseph et al. in [23], where a formalisation of the notion of deductive coherence for graded logics is proposed. In particular, Joseph et al. propose to model deductive coherence problems as graphs: nodes represent graded propositional formulas; edges are the positive or negative constraints between these formulas; and each constraint has a weight expressing the strength of the coherence or incoherence relationship. In this section we apply this formalisation of deductive coherence to consider the relationships among cognitive elements and the mental representation of norms and instances in order to resolve normative inconsistencies.

4.1 Building the Coherence Graph

The coherence graph of an n-BDI agent is defined as follows.

Definition 7 (Coherence Graph [23]) A coherence graph is an edge-weighted undirected graph $g = \langle \mathcal{V}, \mathcal{C}, \zeta \rangle$ where:

- $\mathcal{V}$ is a finite set of nodes representing all the information relevant to the inconsistency problem;
- $\mathcal{C} \subseteq [\mathcal{V}]^2$ is a finite set of constraints representing the coherence or incoherence between nodes[8];
- $\zeta : \mathcal{C} \rightarrow [-1, 1] \setminus \{0\}$ is the coherence function that assigns a value to the coherence between nodes.

Definition 8 (Satisfied Constraints [23]) Given a coherence graph $g = \langle \mathcal{V}, \mathcal{C}, \zeta \rangle$ and a partition $\mathcal{A}$ of $\mathcal{V}$, the set of satisfied constraints $\mathcal{C}_{\mathcal{A}} \subseteq \mathcal{C}$ is:

$$\mathcal{C}_{\mathcal{A}} = \{\{v, w\} \in \mathcal{C} \mid v \in \mathcal{A} \text{ iff } w \in \mathcal{A}, \text{ when } \zeta(\{v, w\}) > 0\} \cup \{\{v, w\} \in \mathcal{C} \mid v \in \mathcal{A} \text{ iff } w \notin \mathcal{A}, \text{ when } \zeta(\{v, w\}) < 0\}$$

All other constraints are said to be unsatisfied.

Definition 9 (Coherence Maximisation [23]) Given a coherence graph $g = \langle \mathcal{V}, \mathcal{C}, \zeta \rangle$, maximising the coherence is the problem of partitioning nodes into two sets (accepted $\mathcal{A}$ and rejected $\mathcal{R} = \mathcal{V} \setminus \mathcal{A}$) such that the coherence of the partition is maximal.

4.1.1 Nodes of the Coherence Graph

In this paper we propose that n-BDI agents resolve inconsistencies by considering: (i) the beliefs that support the activation and expiration of instances (i.e., the formulas in B); (ii) the norms that have been instantiated (i.e., the formulas in N[9]); (iii) instances and the satisfaction and inconsistency relationships between them (i.e., the formulas in N_I); and (iv) the desires that are hindered or favoured by instances (i.e., the formulas in D)[10].

[8] Recall that deductive coherence is a symmetric relationship and, as a consequence, constraints in the coherence graph are defined over the set of all subsets of two elements of $\mathcal{V}$.

[9] Recall that the expressions in N contain a norm and the salience of this norm.

[10] Recall that n-BDI agents translate instances into desires that will be considered for deriving intentions. Thus, intentions are not a basic attitude and there is not a direct link between instances and intentions. As a consequence, the set of intentions is not considered for resolving inconsistencies.

Definition 10 (Nodes of the Coherence Graph) Given an n-BDI agent $\langle B, D, I, N, N_I \rangle$, the nodes of the coherence graph corresponding to this agent are defined as $\mathcal{V} = \mathcal{V}_{N_I} \cup \mathcal{V}_N \cup \mathcal{V}_B \cup \mathcal{V}_D$ where:

- $\mathcal{V}_{N_I} = N_I$
- $\mathcal{V}_N = \left\{ norm(n, \rho_s) \,\middle|\, \begin{array}{c} norm(n, \rho_s) \in N \wedge instance(i, \rho_c) \in N_I : \\ \text{there is a substitution } \sigma \text{ such that } i \text{ is an instance of } n \text{ under } \sigma \end{array} \right\}$
- $\mathcal{V}_B = \left\{ belief(X, \rho_X) \,\middle|\, \begin{array}{c} belief(X, \rho_X) \in B \wedge instance(\langle \Delta, \bar{C}, T, \bar{A}, \bar{E} \rangle, \rho_c) \in N_I : \\ ((X \vdash \bar{A}) \vee (X \vdash play(self, T)) \vee (X \vdash \bar{E})) \end{array} \right\}$
- $\mathcal{V}_D = \left\{ desire(X, \rho_X) \,\middle|\, \begin{array}{c} desire(X, \rho_X) \in D \wedge instance(\langle \Delta, \bar{C}, T, \bar{A}, \bar{E} \rangle, \rho_c) \in N_I : \\ ((X \vdash \bar{C}) \vee (\bar{C} \vdash X)) \end{array} \right\}$

recall that the expression $play(self, T)$ represents that the agent plays role T.

4.1.2 Coherence of the Graph

Now that the nodes of the coherence graph have been defined, the coherence function must be defined. To make sure that the coherence function is symmetric, we first define a general support function [23] between two nodes Φ and Ψ of the coherence graph. This general support function evaluates the support of a node Φ in deriving node Ψ. According to principle **2(a)** of deductive coherence, any proposition coheres with propositions that are deductible from it. In the case of resolving normative inconsistencies, we need to establish coherence relationships between instances and the formulas that deduce them: i.e., the general support function should define the coherence degree between a norm and its instances, between beliefs that support an instance (i.e., beliefs that support the activation of the instance and addressing beliefs) and this instance, between instances that satisfy each other, and between a desire and an instance that is satisfied by the desire. Principle **2(b)** of deductive coherence states that propositions that together are used for deducing some other proposition cohere. Thus, the general support function should also define the coherence degree between beliefs that deduce instances, and between a norm and the beliefs that support instances of this norm. Principle **4** of deductive coherence, which states that contradictory propositions are incoherent, thus the general support function should also define the incoherence degree between an instance and the beliefs that sustain its expiration, between inconsistent instances, and between an instance and the desire hindered by this instance. Finally, note that the aim of the coherence maximization process is not to revise desires or check the coherence of the normative system, and because of this the general support function does not consider other coherence and incoherence relationships.

Given that nodes of the coherence graph might be norms, instances, beliefs and desires, we have defined several specialised support functions according to the type of nodes that are being considered. We formalise the general support function (η) as follows.

Definition 11 (General Support Function for the Coherence Graph) A general support function ($\eta : \mathcal{V} \times \mathcal{V} \rightarrow [-1, 1]$) is given by:

$$\eta(\Phi,\Psi) = \begin{cases} \eta_{BB}(\Phi,\Psi) & \text{if } \Phi = \mathit{belief}(X,\rho_X) \text{ and } \Psi = \mathit{belief}(Y,\rho_Y) \\ \eta_{BN}(\Phi,\Psi) & \text{if } \Phi = \mathit{belief}(X,\rho_X) \text{ and } \Psi = norm(n,\rho_s) \\ \eta_{BN_I}(\Phi,\Psi) & \text{if } \Phi = \mathit{belief}(X,\rho_X) \text{ and } \Psi = instance(i,\rho_c) \\ \eta_{NN_I}(\Phi,\Psi) & \text{if } \Phi = norm(n,\rho_s) \text{ and } \Psi = instance(i,\rho_c) \\ \eta_{N_IN_I}(\Phi,\Psi) & \text{if } \Phi = instance(i^1,\rho_c^1) \text{ and } \Psi = instance(i^2,\rho_c^2) \\ \eta_{N_ID}(\Phi,\Psi) & \text{if } \Phi = instance(i,\rho_c) \text{ and } \Psi = desire(X,\rho_X) \\ \textit{undefined} & \text{otherwise} \end{cases}$$

Support Function for Beliefs (η_{BB}) The specialised support function η_{BB} is responsible for defining the coherence degree between two beliefs ($\mathit{belief}(X,\rho_X)$ and $\mathit{belief}(Y,\rho_Y)$) that are used together for deducing an instance ($instance(\langle \Delta, \bar{C}, T, \bar{A}, \bar{E}\rangle, \rho_c)$). One of these beliefs is the addressing belief and the other is a belief that supports the activation of the instance[11]. η_{BB} has been defined capturing principle **2(b)** of deductive coherence, which claims that those formulas that are used together for deducing some other formula cohere with each other. As a consequence, we define the degree of coherence between these two beliefs as the conjunction between the ease of compliance of the instance that is inferred and the conjunction between the two supporting beliefs. Continuous t-norms are possible truth functions of conjunction in many-valued logics. The three most important continuous t-norms are [18]: Lukasiewicz t-norm, Gödel t-norm and Product t-norm. We have selected the Lukasiewicz t-norm since recent papers on representing and reasoning about graded mental formulas have used it showing successful/interesting results and properties [6]. Thus, we have calculated the conjunction between graded formulas as the strong conjunction according to the Lukasiewicz t-norm [22] as:

$$x \otimes y = max(0, x + y - 1)$$

Thus, we calculate the degree of coherence between the two beliefs considered by the support function η_{BB} as:

$$\rho_c \otimes (\rho_X \otimes \rho_Y) = max(0, \rho_c + max(0, \rho_X + \rho_Y - 1) - 1)$$

where ρ_c is the ease of compliance of the instances inferred from the two beliefs and ρ_X, ρ_Y is the certainty of these beliefs. Principle **2(c)** of deductive coherence claims that the more hypotheses it takes to deduce something, the less the degree of coherence. Accordingly, in function η_{BB} the coherence degree is inversely proportional to the number of formulas that are required to infer instances. As a consequence, the degree of coherence has been divided by 3, since 3 formulas (the activation belief, the addressing belief and the norm) are required to infer the instance. The two beliefs considered by this function can be used

[11] Notice that we assume that the agent performs a reasoning process, such as the one described in [6], for inferring mental formulas (e.g., $\mathit{belief}(a \wedge b, min\{\rho_a, \rho_b\})$) that are a conjunction of separate mental formulas (e.g., $\mathit{belief}(a, \rho_a)$ and $\mathit{belief}(b, \rho_b)$)

together to deduce more than one instance. Thus, we define the coherence between the two beliefs as the maximum among the coherence values caused by each instance that can be inferred from the two formulas. Function η_{BB} is formally defined as follows.

Definition 12 (Support Function for Beliefs) A support function for beliefs ($\eta_{BB} : \mathcal{V} \times \mathcal{V} \rightarrow [0,1] \cup \{undefined\}$) is given by:

$$\eta_{BB}(belief(X,\rho_X), belief(Y,\rho_Y)) =$$
$$\begin{cases} max\left\{ \dfrac{max(0,\rho_c + max(0,\rho_X+\rho_Y-1)-1)}{3} \;\middle|\; \begin{array}{l} instance(\langle \Delta,\bar{C},T,\bar{A},\bar{E}\rangle,\rho_c) \in N_I : \\ (X \vdash \bar{A}) \wedge (Y \vdash play(self,T)) \end{array} \right\} \\ undefined \text{ otherwise} \end{cases}$$

Support Function for Belief and Norms (η_{BN}) Function η_{BN} calculates the coherence degree between a belief and a norm that are used together to deduce an instance. Thus, it has been defined capturing principles **2(b)** and **2(c)** of deductive coherence. Therefore, the coherence degree is calculated as in the previous support function for beliefs. The only difference is that in this case the coherence degree is divided by 3 or 2, depending on the activation condition; if the activation condition is undefined, only two hypotheses are required to infer the instance. Again, the same belief and norm can be used for inferring more than one instance and we define the coherence as the maximum among the coherence values caused by each instance.

Definition 13 (Support Function for Beliefs and Norms) A support function for beliefs and norms ($\eta_{BN} : \mathcal{V} \times \mathcal{V} \rightarrow [0,1] \cup \{undefined\}$) is given by:

$$\eta_{BN}(belief(X,\rho_X), norm(\langle \Delta,C,T,A,E\rangle,\rho_s)) =$$
$$\begin{cases} max\left\{ \begin{array}{l} max\left\{ \dfrac{max(0,\rho_c + max(0,\rho_s+\rho_X-1)-1)}{3} \;\middle|\; \begin{array}{l} instance(\langle \Delta,\bar{C},T,\bar{A},\bar{E}\rangle,\rho_c) \in N_I : \\ A, \bar{A} \text{ are not } \textit{undefined} \text{ and} \\ \text{there is a substitution } \sigma \text{ such that} \\ \langle \Delta,\bar{C},T,\bar{A},\bar{E}\rangle \text{ is an instance of} \\ \langle \Delta,C,T,A,E\rangle \text{ under } \sigma \text{ and} \\ ((X \vdash \bar{A}) \vee (X \vdash play(self,T))) \end{array} \right\} \\ max\left\{ \dfrac{max(0,\rho_c + max(0,\rho_s+\rho_X-1)-1)}{2} \;\middle|\; \begin{array}{l} instance(\langle \Delta,\bar{C},T,\bar{A},\bar{E}\rangle,\rho_c) \in N_I : \\ A, \bar{A} \text{ are } \textit{undefined} \text{ and } ((\Delta = \Delta) \wedge \\ (C = \bar{C}) \wedge (E = \bar{E}) \wedge \\ (X \vdash play(self,T))) \end{array} \right\} \end{array} \right\} \\ undefined \text{ otherwise} \end{cases}$$

Support Function for Beliefs and Instances (η_{BN_I}) This support function defines the coherence among beliefs that support the activation or expiration of instances. Therefore, this function is defined according to principles **2(a)**, **2(c)** and **4** of deductive coherence. Principle **2(a)** claims that any proposition coheres with propositions that are deductible from it. As a consequence, beliefs that deduce instances cohere with these instances. We define the coherence degree between the instance and the belief as the conjunction between these two formulas divided by the number of formulas that are required to infer the instance (according to principle **2(c)**). In the case of a belief that supports the expiration of an instance, this formula is contradictory to the instance[12]. Principle **4** claims that contradictory propositions are incoherent with each other. For this reason, function η_{BN_I} defines incoherence relationships between instances and the beliefs that support the expiration of these instances. Specifically,

[12] Note that agents are still under the influence of any instance even if they stop enacting the target role of this instance. Because of this, we have not defined an incoherence relationship between instances and beliefs that represent the fact that the agent is no longer playing the target role of instances.

we have calculated the incoherence as the weak conjunction according to the Lukasiewicz t-norm [22]:

$$x \wedge y = min(x, y)$$

Thus, we calculate the degree of incoherence between an instance and the belief that supports the instance expiration as:

$$\rho_X \wedge \rho_c = -min(\rho_X, \rho_c)$$

where ρ_c is the ease of compliance of the instance and ρ_X is the certainty of the expiration belief. In this case we use the weak conjunction since an instance and a belief that sustain the expiration of this instance are always incoherent, whatever their degrees.

Definition 14 (Support Function for Beliefs and Instances) A support function for beliefs and instances ($\eta_{BN_I} : \mathcal{V} \times \mathcal{V} \rightarrow [-1, 1] \cup \{undefined\}$) is given by:

$$\eta_{BN_I}(belief(X, \rho_X), instance(\langle \Delta, \bar{C}, T, \bar{A}, \bar{E} \rangle, \rho_c)) =$$
$$\begin{cases} \frac{max(0, \rho_X + \rho_c - 1)}{3} & \text{if } \bar{A} \text{ is not } \textit{undefined and } ((X \vdash \bar{A}) \vee (X \vdash play(self, T))) \\ \frac{max(0, \rho_X + \rho_c - 1)}{2} & \text{if } \bar{A} \text{is } \textit{undefined and } X \vdash play(self, T) \\ -min(\rho_X, \rho_c) & \text{if } X \vdash \bar{E} \\ \textit{undefined} & \text{otherwise} \end{cases}$$

Support Function for Norms and Instances (η_{NN_I}) This support function defines the coherence among norms and their instances as the conjunction between these two formulas divided by the number of formulas that are required to infer the instance.

Definition 15 (Support Function for Norms and Instances) A support function for norm and instances ($\eta_{NN_I} : \mathcal{V} \times \mathcal{V} \rightarrow [0, 1] \cup \{undefined\}$) is given by:

$$\eta_{NN_I}(norm(n, \rho_s), instance(\langle \Delta, \bar{C}, T, \bar{A}, \bar{E} \rangle, \rho_c)) =$$
$$\begin{cases} \frac{max(0, \rho_s + \rho_c - 1)}{3} & \text{if } \bar{A} \text{ is not } \textit{undefined} \text{ and there is a substitution } \sigma \text{ such that} \\ & \langle \Delta, \bar{C}, T, \bar{A}, \bar{E} \rangle \text{ is an instance of } n \text{ under } \sigma \\ \frac{max(0, \rho_s + \rho_c - 1)}{2} & \text{if } \bar{A} \text{ is } \textit{undefined} \text{ and } \langle \Delta, \bar{C}, T, \bar{A}, \bar{E} \rangle \text{ is an instance of } n \text{ under } \emptyset \\ \textit{undefined} & \text{otherwise} \end{cases}$$

Support Function for Function for Instances ($\eta_{N_I N_I}$) This support function creates incoherence links between instances when they are inconsistent. In addition, this support function creates coherence links between instances when they are consistent.

Definition 16 (Support Function for Instances) A support function for instances ($\eta_{N_I N_I} : \mathcal{V} \times \mathcal{V} \rightarrow [-1, 1] \cup \{undefined\}$) is given by:

$$\eta_{N_I N_I}(instance(i^1, \rho_c^1), instance(i^2, \rho_c^2)) = \begin{cases} -min(\rho_c^1, \rho_c^2) & \text{if } i^1 \text{ and } i^2 \text{ are inconsistent} \\ (max(0, \rho_c^1 + \rho_c^2 - 1)) & \text{if } i^1 \text{ satisfies } i^2 \\ \textit{undefined} & \text{otherwise} \end{cases}$$

Support Function for Instances and Desires ($\eta_{N_I D}$) This support function is responsible for creating coherence (vs. incoherence) links between instances and the desires that are favoured (vs. hindered) by them. For example, a desire to achieve a given goal is favoured by an instance that specifies this goal as obligatory. Similarly, a desire to achieve a given goal is hindered by an instance that specifies this goal as forbidden. Thus, $\eta_{N_I D}$ has been defined according to principles **2(a)** and **4** of deductive coherence.

Definition 17 (Support Function for Instances and Desires) A support function for instances and desires ($\eta_{N_I D} : \mathcal{V} \times \mathcal{V} \rightarrow [-1, 1] \cup \{undefined\}$) is given by:

$$\eta_{N_I D}(instance(\langle \Delta, \bar{C}, T, \bar{A}, \bar{E} \rangle, \rho_c), desire(X, \rho_X)) =$$
$$\begin{cases} max(0, \rho_c + \rho_X - 1) & \text{if } \Delta = \mathcal{O} \wedge ((X \vdash \bar{C}) \vee (\bar{C} \vdash X)) \\ -min(\rho_c, \rho_X) & \text{if } \Delta = \mathcal{F} \wedge ((X \vdash \bar{C}) \vee (\bar{C} \vdash X)) \\ \textit{undefined} & \text{otherwise} \end{cases}$$

In this paper we assume that permissions specify exceptions to the application of more general obligations and prohibitions. Therefore, permissions do not directly affect the agent's desires; i.e., they do not imply that some state of affairs must be achieved or avoided. For this reason, we do not create a coherence link between permission instances and desires.

In order to make coherence a symmetric relationship, the deductive coherence between two nodes Φ and Ψ is defined as the maximum of the two support function values that evaluate the support of Φ in deriving Ψ and vice versa. The maximum is chosen due to the fact that, even if there is only a deduction relation in one of the directions, there is a deductive coherence between the two formulas [23].

Definition 18 (Coherence Function[23]) A deductive coherence function for the coherence graph $\zeta : \mathcal{V}^{(2)} \rightarrow [-1, 1] \setminus \{0\} \cup \{undefined\}$ is given by:

$$\zeta(\{\Phi, \Psi\}) = \begin{cases} max(\eta(\Phi, \Psi), \eta(\Psi, \Phi)) & \text{if } \eta(\Phi, \Psi) \neq 0 \text{ and } \eta(\Psi, \Phi) \neq 0 \\ \eta(\Phi, \Psi) & \text{if } \eta(\Phi, \Psi) \neq 0 \text{ and } (\eta(\Psi, \Phi) = 0 \text{ or undefined}) \\ \eta(\Psi, \Phi) & \text{if } \eta(\Psi, \Phi) \neq 0 \text{ and } (\eta(\Phi, \Psi) = 0 \text{ or undefined}) \\ \textit{undefined} & \text{if } \eta(\Psi, \Phi) = 0 \text{ or undefined and } \eta(\Phi, \Psi) = 0 \text{ or undefined} \end{cases}$$

4.1.3 Edges of the Coherence Graph

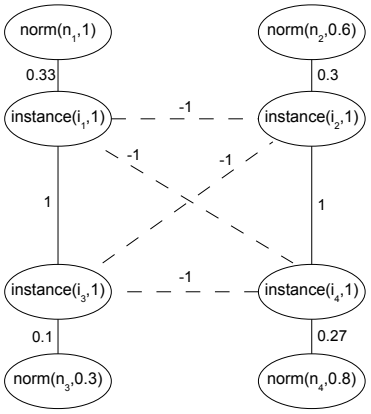
Finally, the set of edges or constraints of a coherence graph are defined as follows

$$\mathcal{C} = \{\{\Phi, \Psi\} | \Phi, \Psi \in \mathcal{V} \text{ and } \zeta(\{\Phi, \Psi\}) \neq \textit{undefined}\}$$

The set of satisfied constraints ($\mathcal{C}_{\mathcal{A}} \subseteq \mathcal{C}$) is calculated as in Definition 8.

The coherence graph corresponding to our running example is depicted by Figure 2a, in which the coherence between the norm n_1 and its instance i_1 is calculated as:

$$\zeta(\{norm(n_1, 1), instance(i_1, 1)\}) = \eta_{NN_I}(norm(n_1, 1), instance(i_1, 1)) = \frac{max(0, 1 + 1 - 1)}{3} = 0.33$$



(a) Consistency Graph

Partition $i_1\ i_2\ i_3\ i_4$	Calculation Method $\sum \lvert\zeta(\{v,w\})\rvert$	$\sum \frac{\lvert\zeta(\{v,w\})\rvert}{\lvert\mathcal{A}\rvert}$
$\surd$	2.27	$\frac{2.27}{2} = 1.14$
$\surd$	2.1	$\frac{2.1}{2} = 1.05$
$\surd\ \surd$	2.37	$\frac{2.37}{4} = 0.59$
$\surd$	2.30	$\frac{2.30}{2} = 1.15$
$\surd\ \surd$	**5.57**	$\mathbf{\frac{5.57}{4} = 1.39}$
$\surd\ \surd$	2.4	$\frac{2.4}{4} = 0.6$
$\surd\ \surd\ \surd$	3.67	$\frac{3.67}{6} = 0.61$
$\surd$	2.33	$\frac{2.33}{2} = 1.17$
$\surd\ \surd$	2.6	$\frac{2.6}{4} = 0.65$
$\surd\ \surd$	5.43	$\frac{5.43}{4} = 1.36$
$\surd\ \surd\ \surd$	3.70	$\frac{3.70}{6} = 0.62$
$\surd\ \surd$	2.63	$\frac{2.63}{4} = 0.66$
$\surd\ \surd\ \surd$	3.90	$\frac{3.90}{6} = 0.65$
$\surd\ \surd\ \surd$	3.73	$\frac{3.73}{6} = 0.62$
$\surd\ \surd\ \surd\ \surd$	3	$\frac{3}{8} = 0.38$

(b) Coherence of partitions

Fig. 2: Inconsistency Resolution in the *webManager* agent example

Similarly, the coherence between the rest of the norms and instances is calculated by the η_{NN_I} function as the conjunction between these two formulas divided by 3, since the activation condition of all norms has been defined (see row N in Table 1). The coherence relationship between instances is calculated by function $\eta_{N_I N_I}$. For example, the coherence between the two instances that forbid use of $fast$ (i.e., the instances i_2 and i_4) is calculated as:

$$\zeta(\{instance(i_2,1), instance(i_4,1)\}) = \eta_{N_I N_I}(instance(i_2,1), instance(i_4,1)) = max(0, 1+1-1) = 1$$

The coherence between the instances i_1 and i_3 is calculated in the same way. Finally, the incoherence between instances is also calculated according to the $\eta_{N_I N_I}$ function. For example, the incoherence between i_1 and i_2 is calculated as:

$$\zeta(\{instance(i_1,1), instance(i_2,1)\}) = \eta_{N_I N_I}(instance(i_1,1), instance(i_2,1)) = -min(1,1) = -1$$

Once the links of the coherence graph have been calculated, it is necessary to calculate the coherence of the partitions to find a partition that maximises coherence.

4.2 Calculating the Coherence Of Partitions

As mentioned in Section 3, Thagard defines the coherence of a partition as the sum of the weights of the constraints satisfied by this partition. According to this definition, two partitions in which the sum of satisfied constraints is the same are equally coherent regardless

of the number of nodes that belong to each partition. In this paper we want to explore alternative definitions for the coherence of a partition. Specifically, we consider the following calculation methods: *sum of satisfied constraints* and *sum of satisfied constraints divided by the cardinality of the accepted set*.

Sum of Satisfied Constraints This calculation method corresponds to the original calculation proposed by Thagard.

Definition 19 (Sum of Satisfied Constraints [39]) Given a coherence graph $g = \langle \mathcal{V}, \mathcal{C}, \zeta \rangle$ the coherence of a partition of nodes in $\mathcal{V}$ into accepted $\mathcal{A}$ and rejected $\mathcal{R}$ sets is calculated as:

$$\sum_{\{v,w\} \in \mathcal{C}_{\mathcal{A}}} |\zeta(\{v, w\})|$$

This is a cumulative measure; i.e., the consideration of new constraints that are satisfied by a partition always increases the coherence of this partition.

Sum of Satisfied Constraints Divided by the Cardinality of the Accepted Set The number of nodes (i.e., instances) that are accepted is an important factor in our problem, since the higher the number of instances that the agent tries to fulfil, the more difficult it may be for the agent to fulfil all of them. Thus, we propose to calculate the coherence of a partition by taking into account the cardinality of the accepted set as follows.

Definition 20 (Sum of Satisfied Constraints Divided by the Cardinality of the Accepted Set) Given a coherence graph $g = \langle \mathcal{V}, \mathcal{C}, \zeta \rangle$, the coherence of a partition of nodes in $\mathcal{V}$ into accepted $\mathcal{A}$ and rejected $\mathcal{R}$ sets is calculated as:

$$\sum_{\{v,w\} \in \mathcal{C}_{\mathcal{A}}} \frac{|\zeta(\{v, w\})|}{|\mathcal{A}|}$$

This is also a cumulative measure.

Other calculation methods that are not cumulative, such as calculating coherence as the sum of satisfied constraints divided by the number of satisfied constraints, have not been considered since, with such methods, the addition of new constraints satisfied by the partition does not always imply that the coherence of this partition increases. This may cause undesirable results. For example, considering a new constraint that supports an instance (e.g., the coherence between an instance and a desire that is favoured by this instance) may decrease the coherence of the partition that contains this instance if the weight of the new constraint is not high enough. This makes no sense, since having more evidence sustaining a particular instance always implies that accepting this instance is more coherent.

4.3 Determining the Partitions for Consideration

Our aim is to dynamically derive a preference order over the power set of active instances, so that for each (non empty) subset of the active instances we construct a partition that contains all elements (cognitive and normative) except those instances that are not part of the subset.

Definition 21 (Partitions for Consideration) Given an n-BDI agent $\langle B, D, I, N, N_I \rangle$ with coherence graph $g = \langle \mathcal{V}, \mathcal{C}, \zeta \rangle$, the partitions of $\mathcal{V}$ that the agent considers are each members of the following set:

$$\{\mathcal{A} \cup \mathcal{V}' \mid \mathcal{A} \in \wp(N_I) \setminus \emptyset \text{ and } \mathcal{V}' = \mathcal{V} \setminus N_I\}$$

For each partition the agent considers, it uses its chosen calculation method (from the previous section) to determine the coherence; from this, the most preferred subset of instances can be derived (i.e., the one that corresponds to the partition that maximises coherence). It is possible that this most preferred subset of instances can contain inconsistent instances; this implies that there is insufficient information present in the coherence graph to decide which of the inconsistent instances should prevail.

As previously mentioned, the problem of solving coherence is NP-complete and the cost of an exhaustive search algorithm increases exponentially with the number of elements. In the case of using coherence to resolve normative inconsistencies, only instances can be accepted or rejected. Thus, the number of possible acceptance sets is $2^{|N_I|}$ which is considerably lower than the $2^{|\mathcal{V}|}$. This means that even for large problems in which an agent has thousands of cognitive and normative elements, our coherence-based approach for resolving inconsistencies can be undertaken using connectionist, semidefinite programming or, even, exhaustive search algorithms since only the specific instances that are active at a given moment are considered as elements that can be accepted or rejected. Moreover, our approach can also be used by bounded rational agents that have limited capabilities for resolving normative inconsistencies. In this case, incremental or greedy algorithms can be used to resolve inconsistencies.

In the running example, there are 15 possible subsets of instances that the *webManager* agent must consider; thus the agent calculates the coherence of the 15 partitions of its coherence graph that correspond to these subsets, according to one of the calculation methods proposed in the previous section. For example, if the *webManager* agent calculates coherence as the sum of satisfied constraints, the coherence of the partition that corresponds to the subset containing instances i_1 and i_3 is calculated as:

$$\sum_{\{v,w\} \in \mathcal{C}_\mathcal{A}} |\zeta(\{v,w\})| = |0.33| + |1| + |0.1| + |-1| + |-1| + |-1| + |-1| = 5.43$$

Similarly, if the *webManager* agent calculates coherence as the sum of satisfied constraints divided by the cardinality of the accepted set, the coherence of the partition that corresponds to the subset containing instances i_1 and i_3 is calculated as:

$$\sum_{\{v,w\} \in \mathcal{C}_\mathcal{A}} \frac{|\zeta(\{v,w\})|}{|\mathcal{A}|} = \frac{5.43}{4} = 1.36$$

Figure 2b presents the coherence of the 15 partitions that the agent must consider, using each of the two calculation methods. Specifically, each partition has been labelled according to the instances that it contains. Regardless of the calculation method used, the partition that maximises coherence is formed by instances i_2 and i_4. In this example, where the ease of compliance of each instance is the same, coherence resolves the inconsistency by proposing a solution in which more than one instance prevails and the static order is not obeyed (i.e., the selected partition does not include the instance of the most salient norm, n_1). Specifically, the *webManager* agent decides to follow the two prohibition instances and not to use either $slow$ or $fast$ due to the fact that the security of the whole system is compromised.

5 Case Study

This section illustrates the performance of the coherence maximisation approach for resolving inconsistencies between two instances. Even in this simple case with only two instances, it is not suitable to analyse mathematically the conditions under which the coherence maximisation process determines that a particular instance prevails over the other. As more nodes (i.e., beliefs, desires and intentions) are added to the coherence graph, the conditions become more complex and difficult to interpret. For these reasons, we have carried out experiments to illustrate the performance of our coherence maximisation process for resolving inconsistencies between two instances belonging to our running example. These experiments compare our proposal against existing proposals and demonstrate that it allows agents to solve normative inconsistencies by adapting the solution in response to a dynamic, uncertain and non-deterministic environment. In particular, we have compared three inconsistency resolution strategies: (i) maximising *coherence*, which is the method proposed in this paper; (ii) following the *static order* according to salience, which is the method used in the majority of previous proposals [4,7,21] (see Section 7 for a discussion of previous work); and (iii) following a *conditional order* that determines which instance prevails according to a condition [41].

In our experiments, we assume that there are two norms: the norm that forbids university members to use $fast$:

$$n^F = \langle \mathcal{F}, use(fast), universityMember, \top, \bot \rangle$$

and the norm that permits academic staff to use the $fast$ server when $slow$ is overloaded:

$$n^P = \langle \mathcal{P}, use(fast), academicStaff, highTraffic(slow), lowTraffic(slow) \rangle$$

The *webManager* determines that $slow$ has high traffic when it has been accessed more than 1000 times in the last hour. Accordingly, the certainty of the $highTraffic(slow)$ proposition is calculated by considering the number of visits that $slow$ has received in the last hour (which is denoted by v_1) as follows:

$$\begin{cases} 1 & v_1 \geq 1000 \\ \frac{v_1}{1000} & 0 < v_1 < 1000 \end{cases}$$

Similarly, the *webManager* agent determines that $slow$ has low traffic when it has been accessed less than 2500 times in the last 24 hours. Thus, the certainty of the $lowTraffic(slow)$ proposition is calculated by considering the number of visits that $slow$ has received in the last 24 hours (which is denoted by v_{24}) as follows:

$$\begin{cases} 1 & v_{24} = 0 \\ 1 - \frac{v_{24}}{2500} & 0 < v_{24} < 2500 \\ 0 & v_{24} \geq 2500 \end{cases}$$

For example, when $slow$ has received 1500 visits during the last hour and 5000 visits in the last 24 hours, it is considered as high traffic with a certainty of 1. In contrast, when $slow$ has received 0 visits during the last hour and 50 visits in the last 24 hours, it is considered as low traffic with a certainty of 0.98. Moreover, $slow$ can have neither high traffic nor low traffic. For example, when $slow$ has received 0 visits during the last hour and 3000 visits in the last 24 hours, it cannot be considered as high traffic or low traffic. Finally, $slow$ can

have both high and low traffic simultaneously. Suppose $slow$ has received 280 visits during the last hour and 1500 visits in the last 24 hours, this can be considered as high traffic with a certainty of 0.28 and as low traffic with a certainty of 0.4.

Thus our simulation allows us to consider situations in which agents have to resolve inconsistencies among two instances according to uncertain or conflicting data. In our experiments, we compute: the percentage of times that the *permission* instance prevails; the percentage of times that the *prohibition* instance prevails; and the percentage of times that the inconsistency remains *unresolved*. In the situations where the inconsistency remains unresolved, the agent does not have sufficient information to decide between the two instances and it is better to postpone the decision.

5.1 Influence of the Normative Elements on the Resolution of Inconsistencies

In this section we describe the results of an experiment that illustrates how the normative elements present in the agent's knowledge base affect the resolution of the inconsistency. Let us consider the case in which a website that has been created by a lecturer (a member of academic staff) was highly accessed in the past. Thus, two inconsistent instances were created out of the two norms as follows:

$$i^{\mathcal{F}} = \langle \mathcal{F}, use(fast), universityMember, \top, \bot \rangle$$

$$i^{\mathcal{P}} = \langle \mathcal{P}, use(fast), academicStaff, highTraffic(slow), lowTraffic(slow) \rangle$$

Sometime later, this lecturer leaves the university. As a result, the access to his website decreases. Specifically, the website has received 0 visits during the last hour and 3000 visits in the last 24 hours. This website cannot be considered either as a low-traffic website or a high-traffic website (since $f_{highTraffic}(0) = 0$ and $f_{lowTraffic}(3000) = 0$), thus the agent has no current beliefs that entail either the activation or expiration of either instance. This scenario is depicted in Figure 3.

The *webManager* agent must resolve the inconsistency between the two instances by considering the ease of compliance of the instances and the salience of the norms that have given rise to the instances. When the permission norm is the most salient and the permission instance is the easiest to comply with, it seems reasonable that the *webManager* ignores the prohibition instance and follows the permission. Similarly, when the prohibition norm is the most salient and the prohibition instance is the easiest to comply with, then the *webManager* must focus on complying with the prohibition instance. When one norm is more important but the agent is less able to comply with its instance, then it is not obvious which instance should be followed. However, the permission norm is instantiated under certain conditions that cannot be verified in this scenario (i.e., in the scenario depicted by Figure 3 there is no belief supporting the activation of the permission instance, since the website has not received a visit in the last hour). In contrast, the prohibition norm is instantiated by default (i.e., its activation condition is undefined). When the ease of compliance order is inconsistent with the salience order, and in the absence of any evidence that supports the activation of the permission norm, it is reasonable to prefer the prohibition instance, since it is always active and its validity can be taken for granted.

To examine how the coherence maximisation approach performs in such a situation (where there are two inconsistent instances and no cognitive elements relevant to either instance or to the norms they instantiate), we have performed an experiment that simulates the scenario depicted by Figure 3, considering a range of values for the ease of compliance

of each instance and for the salience of the norms they instantiate (summarised in Table 2). Thus, for each run of the experiment an agent is affected by two inconsistent instances i^P and i^F that have been created out of norms n^P and n^F, respectively; the salience of each norm (i.e., ρ_s^P and ρ_s^F) and the ease of compliance of each instance (i.e., ρ_c^P and ρ_c^F) are each randomly assigned to a real value within the $[0, 1]$ interval. In this case, the conditional order is defined considering the average of the ease of complying with an instance and the salience of the norm it instantiates[13]. In all the experiments, we performed 1000 runs of the experiment.

Parameter	Value
Salience of the Permission Norm (ρ_s^P)	$[0, 1]$
Salience of the Prohibition Norm (ρ_s^F)	$[0, 1]$
Ease of Compliance of the Permission Instance (ρ_c^P)	$[0, 1]$
Ease of Compliance of the Prohibition Instance (ρ_c^F)	$[0, 1]$
Number of simulations	1000

Table 2: Parameters used for simulating an inconsistent pair of instances when there are no cognitive elements relevant to the situation.

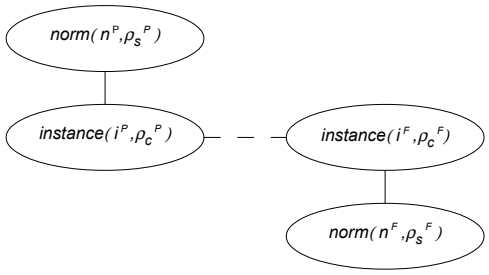


Fig. 3: Coherence graph of an inconsistent pair of instances when there are no cognitive elements relevant to the situation.

Figure 4 depicts the results of this experiment. Specifically, we analyse the resolution of the inconsistency in four different situations: *Situation A*, when the permission norm is the most salient and the permission instance is the easiest to comply with (i.e., when $\rho_s^P \geq \rho_s^F$ and $\rho_c^P \geq \rho_c^F$); *Situation B*, when the permission norm is the most salient but the prohibition instance is the easiest to comply with (i.e., when $\rho_s^P \geq \rho_s^F$ and $\rho_c^P < \rho_c^F$); *Situation C*, when the prohibition norm is the most salient but the permission instance is the easiest to comply with (i.e., when $\rho_s^P < \rho_s^F$ and $\rho_s^P \geq \rho_s^F$); and *Situation D*, when the prohibition norm is the

[13] In particular, the conditional order prefers the prohibition instance to the permission instance iff

$$\frac{\rho_s^F + \rho_c^F}{2} > \frac{\rho_s^P + \rho_c^P}{2}$$

otherwise the permission instance prevails.

most salient and the prohibition instance is the easiest to comply with (i.e., when $\rho_s^P < \rho_s^F$ and $\rho_s^P < \rho_s^F$).

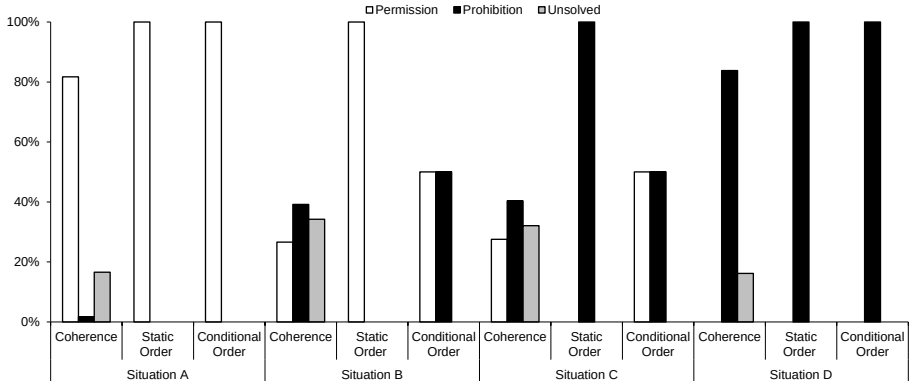


Fig. 4: For each situation, percentage of times in which the permission instance is followed, percentage of times in which the prohibition instance is followed and percentage of times the inconsistency remains unresolved for each resolution strategy. In Situation A $\rho_s^P \geq \rho_s^F$ and $\rho_c^P \geq \rho_c^F$. In Situation B $\rho_s^P \geq \rho_s^F$ and $\rho_c^P < \rho_c^F$. In Situation C $\rho_s^P < \rho_s^F$ and $\rho_c^P \geq \rho_c^F$. In Situation D $\rho_s^P < \rho_s^F$ and $\rho_c^P < \rho_c^F$.

Situations A and D each represent a situation in which the instance that it is easiest to comply with is created from the most salient norm. As Figure 4 depicts, in these situations the same results are obtained if the agent follows either the static or the conditional order; in each case, the instance that has been created out of the most salient norm always prevails. If we consider, however, the performance of the coherence approach in these situations, we see that in 17% of runs the inconsistency remained unresolved. These unresolved inconsistencies correspond to situations in which the agent does not have enough information to make a decision. For example, if the ease of compliance of the instances is very low, which means that the agent will find it hard to fulfil any of the instances, then it may be better to postpone the decision to a later moment at which it has a greater chance of fulfilling the instances.

Situations B and C each represent a controversial situation in which the instance that is easiest to comply with has been created out of the least salient norm. In these situations, following the static order is not enough, since it always results in the agent following the instance that has been created out of the most salient norm, regardless of the agent's capability of complying with it. Thus, the agent may decide to abide by an instance that it is unable to comply with. The conditional order determines that either of the two instances is adhered to with a probability of 50%. Finally, coherence is the only strategy that takes into account the fact that the permission instance does not have an activation or expiration condition and so is always active (i.e., the percentage of times that the prohibition instance is followed is slightly higher than the percentage of times that the permission instance is followed). Thus, coherence determines that the prohibition instance prevails in 39% of runs, whereas the permission prevails in 27% of runs. Finally, coherence determines that in 32% of the cases the agent does not have enough information and the inconsistency remains unresolved. Note that in Situations B and C the information available for making a decision about norm com-

pliance is contradictory and, as a consequence, the percentage of unresolved inconsistencies is higher than in Situations A and D.

As previously mentioned there are two calculation methods by which the coherence of partitions can be calculated. Figure 4 shows the results obtained when we calculate the coherence of partitions as the sum of satisfied constraints. We have repeated the experiment described here but instead calculating the coherence of partitions as the sum of satisfied constraints divided by the cardinality of the accepted set. Since only two instances are considered, the results obtained by the two methods are very similar (the only difference is that the percentage of unresolved inconsistencies is slightly lower when the coherence of partitions is calculated as the sum of satisfied constraints divided by the cardinality of the accepted set). Thus, in this section we only include results obtained when coherence is calculated as the sum of satisfied constraints. Section 6 presents the results of an experiment that compares the two calculation methods.

Coherence approach more often avoids instances that cannot be fulfilled. The results shown in Figure 4 demonstrate that our coherence approach produces different results to both the static order and the conditional order. In order to demonstrate the appropriateness of the result produced by the coherence approach, we have performed some calculations to determine how often during our experiment each of the approaches selected an instance to comply with that cannot be fulfilled.

We assume that agents are situated in a non-deterministic environment under uncertainty and, as a consequence, it may be the case that the agent is unable to comply with a particular instance. We determine whether this is the case probabilistically, where the probability in which each instance cannot be fulfilled is 1 minus its ease of compliance[14]. From this figure we determine that:

- in 22.84% of all runs, the coherence approach returns an instance that the agent cannot comply with[15];
- in 49.24% of all runs, the static order approach returns an instance that the agent cannot comply with;
- in 37.52% of all runs, the conditional order approach returns an instance that the agent cannot comply with.

Note that even if the instance that is easiest to comply with is always selected, then in 32.54% of cases the instance that is selected cannot be fulfilled. This analysis shows that the coherence approach is less likely than both the conditional and the static order approaches to return an instance that it is not possible to comply with.

We have also analysed those situations in which each of the two instances can be fulfilled to consider how often each approach selects the least salient norm. We determine that:

- in 5.58% of those runs in which the agent can comply with both instances, the coherence approach returns the instance that has been created from the least salient norm[16];
- in 0% of those runs in which the agent can comply with both instances, the static order approach returns the instance that has been created from the least salient norm;

[14] In each run we generate a random number for each instance. We define that the instance can be fulfilled when this number is greater than 1 minus its ease of compliance.

[15] Note that we only consider the runs in which coherence selects one instance (i.e., the inconsistency does not remain unresolved) and this instance cannot be fulfilled.

[16] Note that we only consider the runs in which both instances can be fulfilled and coherence selects the least salient instance (i.e., the inconsistency does not remain unresolved).

– in 12.54% of those runs in which the agent can comply with both instances, the conditional order approach returns the instance that has been created from the least salient norm.

From this analysis we see that in only a few of those cases where the agent can comply with both instances (5.58%), the coherence approach returns the instance that has been created from the least salient norm. While it might seem more desirable in such cases that the instance that was created from the most salient norm is always the one returned (such as we see with the static order approach), the benefit gained from the coherence approach by returning an instance that an agent cannot comply with significantly less often than the other approaches offsets this behaviour.

5.2 Influence of the Cognitive Elements on the Resolution of Inconsistencies

5.2.1 Taking into Account the Roles Played by Agents

In this experiment we analyse the influence of the addressing beliefs. If the *webManager* knows that its user is an academic, then it seems more reasonable to take into account the permission instance, since it affects only academic staff. Similarly, if the *webManager* only knows that its user is a university member, then it seems more reasonable to take into account the prohibition instance. If the agent does not know if its user is either an academic or a university member[17], then the decision must be taken according to the normative elements (i.e., according to salience and ease of compliance). Moreover, we are interested in resolving inconsistency when the agent has ambiguous beliefs. Thus, if the *webManager* is sure that its user is both a university member and an academic, then the agent knows that the two norms are addressed to its user. In this situation the agent is affected by two inconsistent instances that are sustained by two addressing beliefs that have the maximum certainty. In the previous experiment (described in Section 5.1), the agent was also affected by two inconsistent instances. However, these instances were not supported by any addressing belief, which means that the agent has no evidence supporting the fact that it is its responsibility to fulfil the instances. In the current experiment, the agent is completely sure that it is responsible for fulfilling the two inconsistent instances. Because of this, the agent cannot postpone this decision and should determine which one prevails. The agent is completely sure that the user is affected by the prohibition instance, since the user is enacting the target role and the norm is instantiated by default. In contrast, the agent has less evidence about the fact that its user is affected by the permission (i.e., it only knows that the user is enacting the target role, but it has no belief about the activation condition). Thus, it makes sense to give greater precedence to the prohibition norm, since the agent is completely sure that it affects its user.

To investigate how the coherence maximisation approach performs in the situation where there are two inconsistent instances and beliefs about the addressing role of each instance, we performed an experiment that simulates the scenario shown in Figure 5, considering the range of values for the ease of compliance of each instance and for the salience of the norms they instantiate given in Table 2, and randomly assigning the values of $\rho_{academicStaff}$ and $\rho_{universityMember}$ (i.e., the certainty that the owner of the website is playing the role of academic staff and university member, respectively) to either 0 or 1. In this case, the *conditional order* is defined by considering the average of the ease to comply with an instance,

[17] It may be the case that the two norms were instantiated at two points in the past, when the *webManager* knew that its user was an academic and a university member. However, the *webManager* cannot determine in the current situation whether its user is still the target of the two instances.

the salience of the norm it instantiates and the certainty of the addressing belief that supports it[18].

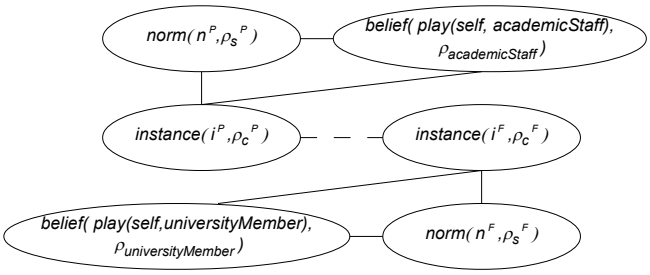


Fig. 5: Resolution of inconsistencies when there are addressing beliefs.

Figure 6 depicts the results. Specifically, we analyse the resolution of the inconsistency in four different situations: *Situation A*, when the *webManager* knows that its user is both an academic and a member of the university (i.e., when both $\rho_{academicStaff}$ and $\rho_{universityMember}$ are 1); *Situation B*, when the *webManager* only knows that its user is an academic (i.e., when $\rho_{academicStaff}$ is 1 and $\rho_{universityMember}$ is 0); *Situation C*, when the *webManager* only knows that its user is a member of the university (i.e., when $\rho_{universityMember}$ is 1 and $\rho_{academicStaff}$ is 0); and *Situation D*, when the *webManager* does not know whether its user is a member of the university or an academic (i.e., when both $\rho_{academicStaff}$ and $\rho_{universityMember}$ are 0).

If the agent follows the static order, then the addressing beliefs and any cognitive elements are not taken into account and the behaviour exhibited by the *webManager* is the same in all situations, so in the following sections we focus our discussion on the difference in behaviour between the conditional order and the coherence approach.

In situations labelled as A the agent is addressed two inconsistent instances with the maximum certainty. If the inconsistency is resolved by means of coherence, then the *webManager* is able to recognise that the prohibition instance is active by default (as it has no activation condition) while the agent has no current belief about the permission's activation condition. Specifically, the coherence approach recognises that the agent has less evidence to support the requirement to fulfil the permission and determines that the prohibition prevails almost twice as often as it determines that the permission prevails. In contrast, when the agent resolves the inconsistency according to the conditional order, it makes the decision according to the salience and the ease of compliance[19] and is equally likely to determine that either the prohibition or the permission instance should prevail.

Situations B and C each represent a situation in which only one of the instances is sustained by an addressing belief (in Situation B, only the permission instance is sustained by an addressing belief; in Situation C, only the prohibition instance is sustained by an addressing

[18] In particular, the conditional order prefers the prohibition instance to the permission instance iff

$$\frac{\rho_s^F + \rho_c^F + \rho_{universityMember}}{3} > \frac{\rho_s^P + \rho_c^P + \rho_{academicStaff}}{3}$$

otherwise the permission instance prevails.

[19] Notice that the two addressing beliefs have a certainty of 1.

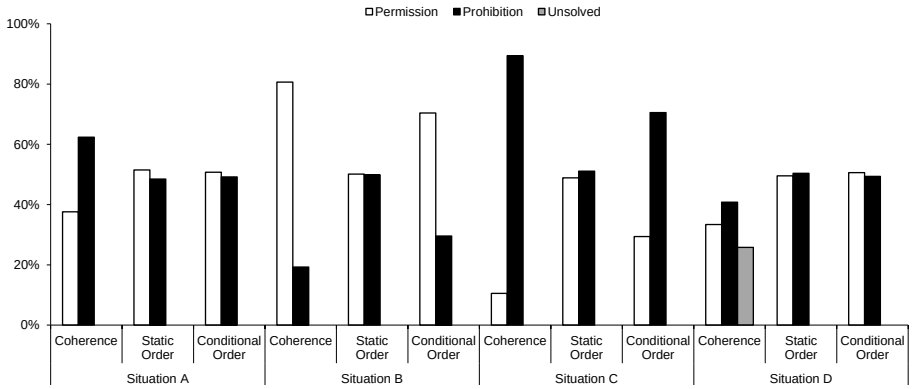


Fig. 6: For each situation, percentage of runs in which the permission instance is followed, percentage of runs in which the prohibition instance is followed and percentage of runs in which the inconsistency remains unresolved for each resolution strategy. In Situation A, $\rho_{academicStaff} = \rho_{universityMember} = 1$. In Situation B, $\rho_{academicStaff} = 1$ and $\rho_{universityMember} = 0$. In Situation C, $\rho_{universityMember} = 1$ and $\rho_{academicStaff} = 0$. In Situation D, $\rho_{academicStaff} = \rho_{universityMember} = 0$.

belief). In these situations, both the conditional order and the coherence approach determine that the instance that is supported by an addressing belief should prevail significantly more often than the instance that is not supported by an addressing belief. In a few cases, each approach does decide that the instance that is not supported by an addressing belief should prevail, when its ease of compliance and/or the salience of the norm it instantiates is higher than that of the instance that is supported by an addressing belief. The main difference between the two approaches is that the coherence approach gives preference to the instances supported by an addressing belief more often than the conditional approach.

In runs that fall under situation D, neither of the instances is sustained by an addressing belief. Situation D is thus the same scenario that we examine in Section 5.1, in which the instances were not supported by any addressing beliefs. For this reason, the results obtained by each approach are the averages of all situations for the results presented in Section 5.1.

The coherence approach is the only resolution strategy that is able to differentiate between Situations A and D. Specifically, in Situation D (where it is unsure the website owner's roles require it to fulfil each of the instances) the coherence approach determines that there is insufficient information to decide which instance should prevail in 26% of situations. This is desirable since in the situation where the agent is not certain that the website owner is required to comply with each instance, it is reasonable that it requires more information to distinguish between the two instances.

Coherence approach more often avoids instances that are not addressed to the agent. To further demonstrate the appropriateness of the result produced by the coherence approach, we have performed calculations to determine how often during our experiment each of the approaches selected an instance to comply with that is not addressed to the agent. Again we assume that agents are situated in a non-deterministic environment under uncertainty and, as a consequence, they cannot always determine correctly if they are the addressee of a given instance. We determine this probabilistically, and specify that agents make mistakes when

they determine the certainty of the addressing beliefs with a probability of 0.1. In particular, we have analysed those runs in which the agent is the addressee of some of the instances (i.e., we focus on situations where the agent is responsible for fulfilling at least one instance) and we determine that:

- in 14.27% of these runs, the coherence approach returns an instance that is not addressed to the agent;
- in 33.27% of these runs, the static order approach returns an instance that is not addressed to the agent;
- in 23.78% of these runs, the conditional order approach returns an instance that is not addressed to the agent.

This analysis shows that the coherence approach is less likely than both the conditional and the static order approaches to return an instance that is not addressed to the agent significantly less frequently than other approaches.

5.2.2 Taking into Account the Agent Environment

We now consider the scenario in which the *webManager* agent has beliefs relating to the activation and the expiration condition of the permission instance. In such a situation, the agent has contradictory beliefs that sustain both the activation and the expiration of the permission instance. According to the norm semantics assumed in this paper[20], the expiration condition specifies when an instance expires and should no longer be considered for compliance.

To investigate how the coherence maximisation approach performs in the situation where there are two inconsistent instances and beliefs about the expiration and the activation of one of these instances, we have performed an experiment that simulates the scenario shown in Figure 7, considering the range of values for the ease of compliance of each instance and for the salience of the norms they instantiate given in Table 2, randomly assigning the value of v_1 (the number of visits received by $slow$ in the last hour) to a natural number in the interval $[0, 2500]$ and randomly assigning the value of v_{24} (the number of visits received by $slow$ in the last 24 hours) to a natural number in the interval $[w_1, 5000]$. In this case, the *conditional order* is defined by considering the average of the ease to comply with an instance, the salience of the norm it instantiates, and (in the case of the permission only) the certainty of the activation belief that supports it and 0 minus the certainty of the belief that supports its expiration[21].

Figure 8 depicts the results of this experiment. Specifically, we analyse the resolution of the inconsistency in four different situations: in *Situation A* the *webManager* has high-certainty beliefs that sustain both the activation and the expiration of the permission instance (i.e., $\rho_{highTraffic(slow)} \geq 0.5$ and $\rho_{lowTraffic(slow)} \geq 0.5$); in *Situation B* the *webManager* has a high-certainty belief that sustains the activation of the permission instance and a low-certainty belief that sustains the expiration of the permission instance (i.e., $\rho_{highTraffic(slow)} \geq 0.5$ and $\rho_{lowTraffic(slow)} < 0.5$); in *Situation C* the *webManager* has a low-certainty belief that sustains the activation of the permission instance and a high-certainty belief that sustains the expiration of the permission instance (i.e., $\rho_{highTraffic(slow)} <$

[20] Such semantics have been widely used in previous research on agents and norms, such as [31] and [27].

[21] The conditional order prefers the prohibition instance to the permission instance iff

$$\frac{\rho_s^F + \rho_c^F}{2} > \frac{\rho_s^P + \rho_c^P + \rho_{highTraffic(slow)} - \rho_{lowTraffic(slow)}}{4}$$

otherwise the permission instance prevails.

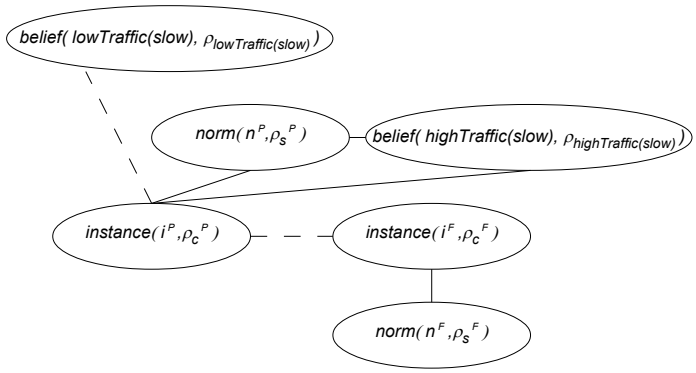


Fig. 7: Resolution of inconsistencies when there are beliefs that support the activation and expiration of instances.

0.5 and $\rho_{lowTraffic(slow)} \geq 0.5$); and in *Situation D* the *webManager* has a low-certainty belief that sustains the activation of the permission instance and a low-certainty belief that sustains the expiration of the permission instance (i.e., $\rho_{highTraffic(slow)} < 0.5$ and $\rho_{lowTraffic(slow)} < 0.5$).

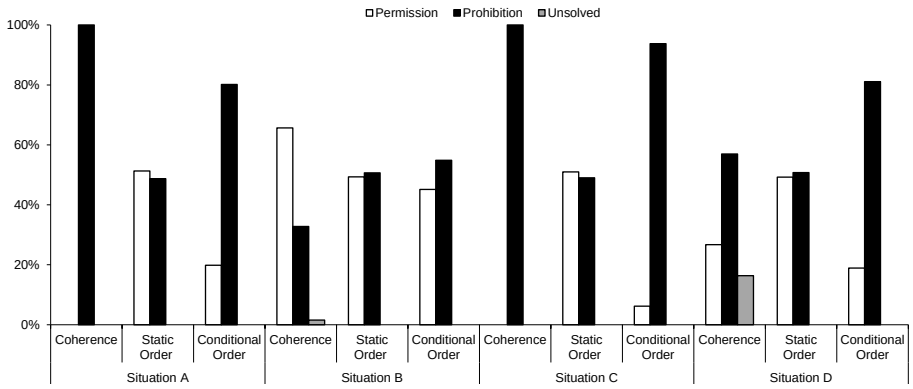


Fig. 8: For each situation, percentage of times in which the permission instance is followed, percentage of times in which the prohibition instance is followed and percentage of times in which the inconsistency remains unresolved for each resolution strategy. In Situation A, $\rho_{highTraffic(slow)} \geq 0.5$ and $\rho_{lowTraffic(slow)} \geq 0.5$. In Situation B, $\rho_{highTraffic(slow)} \geq 0.5$ and $\rho_{lowTraffic(slow)} < 0.5$. In Situation C, $\rho_{highTraffic(slow)} < 0.5$ and $\rho_{lowTraffic(slow)} \geq 0.5$. In Situation D, $\rho_{highTraffic(slow)} < 0.5$ and $\rho_{lowTraffic(slow)} < 0.5$.

Situation A represents those cases where there is a high certainty about both the activation and the expiration of the permission. Here, where there is a high certainty that the permission instance has expired, both the conditional order and the coherence approach determine that the prohibition should prevail in the majority of cases; this is desirable be-

haviour since it is very likely that the permission has expired and so should not be considered. This behaviour is more pronounced with the coherence approach (with which the prohibition prevails in 100% of cases, compared with 80% of cases with the conditional order approach), thus the coherence approach appears to be more heavily influenced by the expiration beliefs than the conditional order.

In Situation B, the *webManager* has a high-certainty belief that sustains the activation of the permission instance and a low-certainty belief that sustains the expiration of the permission instance. Here we see a marked difference in behaviour of the coherence approach and the conditional approach, with coherence determining that the permission should prevail in the majority of cases (66%) while the conditional approach determines that the prohibition should prevail in the majority of cases (55%). The behaviour of the coherence approach in this situation is desirable since the agent has a strong specific belief that it must consider the permission, no strong belief that it ought not to consider the permission, and there are no specific beliefs relating to the prohibition. Thus the coherence approach appears to be more heavily influenced by the activation beliefs than the conditional order.

Situation C corresponds to cases in which the *webManager* has a low-certainty belief that sustains the activation of the permission instance and a high-certainty belief that sustains the expiration of the permission instance. Both the conditional order and the coherence approach determine that the prohibition should prevail in the majority of cases (100% of cases with the coherence approach; 94% of cases with the conditional order approach), as we might expect since in the case in which the permission has expired, the agent ought not to consider it. We argue that the behaviour of the coherence approach is more desirable than that of the conditional order, since it always determines that the prohibition should prevail; this is reasonable since in this case there is high certainty in the expiration belief.

Finally, situation D represents cases where the *webManager* has a low-certainty belief that sustains the activation of the permission instance and a low-certainty belief that sustains the expiration of the permission instance. If the agent resolves the inconsistency by means of coherence, then the results are close to the average of the results obtained by the coherence approach in the first experiment, where there were only normative elements for consideration. In this case we observe that the prevalence of the prohibition is higher than in the first experiment; this is reasonable since we have a low certainty belief that supports the expiration of the permission instance. If the agent resolves the inconsistency according to the conditional order, then the prohibition prevails in almost all cases; this is not a desirable outcome since a low-certainty belief in the expiration of the permission should not be enough to mean that it is almost never considered. In the case where the agent is not very sure that the permission has expired, the other factors for consideration (i.e., salience, ease of compliance, certainty of activation) ought to influence the outcome, as we see with the coherence approach.

Coherence approach more often avoids expired instances. To determine how often during our experiment each of the approaches selected an instance to comply with that has expired, we performed some calculations. In particular, we assume that agents are situated in a non-deterministic environment under uncertainty and, as a consequence, they cannot determine with certainty whether the permission has expired (i.e., if $lowTraffic(slow)$ is true). We determine whether this is the case probabilistically, where the probability that the permission has expired is 1 minus the certainty that $slow$ is experiencing low traffic. In particular, we have analysed those runs in which the permission has expired and we determine that:

- in 8.79% of these runs, the coherence approach returns an instance that has expired (i.e., it returns the permission norm);

– in 51.91% of these runs, the static order approach returns an instance that has expired;
– in 20.99% of these runs, the conditional order approach returns an instance that has expired.

This analysis shows that the coherence approach is less likely than both the conditional and the static order approaches to return an expired instance. This is desirable behaviour since there is no need to comply with expired instances.

5.2.3 Taking into Account the Agent Desires

Here we assume that the website owner has an explicit desire that the website is allocated to $fast$ ($use(fast)$). The desirability degree of this desire ($\rho_{use(fast)}$) indicates how important this desire is to the website owner. For example, a website containing some funny graphic animations might not be important enough to strongly desire that it is allocated to $fast$, even if $slow$ is overloaded and requests are served with long delays; while it may be very important that a website through which students obtain files needed for an assignment is allocated to $fast$. The coherence graph corresponding to this scenario is depicted by Figure 9. The desire to use $fast$ is hindered by the prohibition and so there is a negative constraint between the desire and the prohibition instance.

To investigate how the coherence maximisation approach performs in the situation where there are two inconsistent instances and a desire that is hindered by one of these instances, we performed an experiment that simulates the scenario shown in Figure 9, considering the range of values for the ease of compliance of each instance and for the salience of the norms they instantiate given in Table 2, and randomly assigning the value of $\rho_{use(fast)}$ to a real value in the interval $[0, 1]$. In this case, the *conditional order* is defined by considering the average of the ease to comply with an instance, the salience of the norm it instantiates, and (in the case of the prohibition only) the desirability degree of the desire that it hinders.[22]

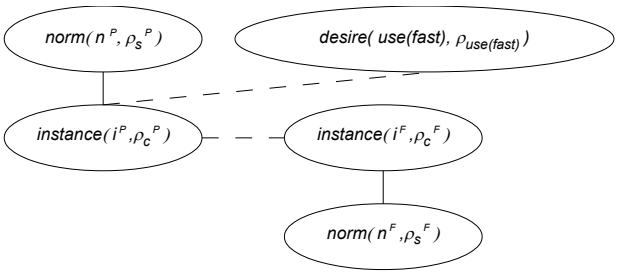


Fig. 9: Resolution of Inconsistencies according to convenience of instances.

Figure 10 depicts the results of this experiment. Specifically, we analyse the resolution of the inconsistency in two different situations: in *Situation A*, the *webManager* strongly

[22] In particular, the conditional order prefers the prohibition instance to the permission instance iff

$$\frac{\rho_s^F + \rho_s^F - \rho_{use(fast)}}{3} > \frac{\rho_s^P + \rho_c^P}{2}$$

otherwise the permission instance prevails.

desires that the website is allocated to the *fast* server (i.e., $\rho_{use(fast)} \geq 0.5$); in *Situation B*, the *webManager* weakly desires that the website is allocated to the *fast* server (i.e., when $\rho_{use(fast)} < 0.5$).

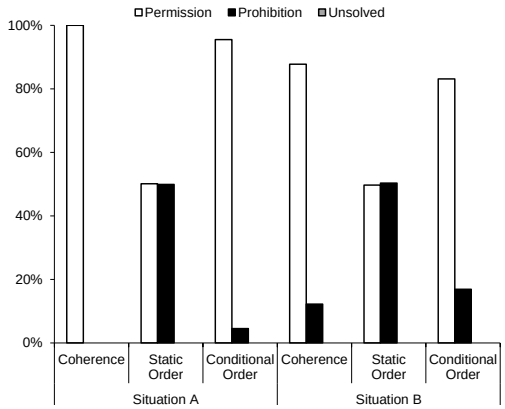


Fig. 10: Percentage of times in which the permission instance is followed, percentage of times in which the prohibition instance is followed and percentage of inconsistencies that remain unresolved for each resolution strategy. In Situation A $\rho_{use(fast)} \geq 0.5$. In Situation B $\rho_{use(fast)} < 0.5$.

We see that in both situations, both the conditional order and the coherence approach determine that the permission should prevail in the majority of cases. This is reasonable, since the addition of a desire that is hindered by the prohibition increases the appropriateness of the permission. Moreover, both approaches determine that the lower the desirability of this desire, the lower the prevalence of the permission norm. However, this decrease is more noticeable if the coherence approach is used; i.e., when coherence is used the prevalence of the permission norm is reduced by 88%, whereas when the conditional order is used the prevalence of the permission norm is reduced by 87%. Thus we see that with the coherence approach, the desire has less of an influence on the decision when it is less strongly desired.

Coherence approach more often avoids undesired instances. We have performed calculations to determine how often during our experiment each of the approaches selected an instance to comply with that is not desired. Again, we assume that agents are situated an non-deterministic environment under uncertainty and, as a consequence, they cannot infer with certainty the interest of a website (i.e., if it is true that the user desires to allocate a web page to the *fast* server). We determine whether this is the case probabilistically, where the probability with which the user wants to allocate a web page to *fast* is the desirability of $use(fast)$. In particular, we analysed those runs in which the user wants to allocate a web page to *fast* and determine that:

- in 1.59% of these runs, the coherence approach returns an instance that is not desired (i.e., it returns the prohibition instance);
- in 48.97% of and runs, the static order approach returns an instance that is not desired;
- in 6.01% of and runs, the conditional order approach returns an instance that is not desired.

This analysis shows that the coherence approach is less likely than both the conditional and the static order approaches to return an undesired instance.

5.3 Discussion of Results

As demonstrated by the experiments presented in this section, coherence allows the online resolution of inconsistencies among instances by taking into account the cognitive and normative elements. Our experiments demonstrate that the coherence approach adapts its decision to the particular situation and obtains more appropriate results than either the static or the conditional order, particularly when it is not obvious from the information that must be taken into account which instance should prevail. As a consequence, agents are able to use the coherence approach to deal with dynamic and non-deterministic environments controlled by norms, such as in the case study described in this section. In particular, our experiments demonstrate that our coherence approach better suited than both the conditional order and the static order approaches when faced with uncertainty in the estimation of the ease of compliance, the certainty of the beliefs and the desirability of desires. Moreover, coherence is a general solution that is not domain dependent and can be applied to scenarios in which agents must resolve inconsistencies according to ambiguous, conflicting and incomplete information. In contrast, approaches that depend on a predefined static order based on the salience of the norms are unable to adapt to the agent's circumstances, and the conditional approach (which takes into account the degrees of all of the relevant normative and cognitive elements) obtains undesirable results in some situations. Specifically, the experiments demonstrate that the conditional order fails to deal well with information that is incompatible with the instances[23].

One of the most important features of coherence is that it allows controversial situations (i.e., situations where the agent does not have enough information to make a decision) to be identified by leaving the inconsistency unresolved; neither the static nor the conditional approach are ever able to produce such a result. Unresolved inconsistencies give agents the opportunity to postpone the decision and reason further. For example, agents may realise that they are playing roles that entail responsibilities that they cannot fulfil and they may be able to leave these roles or even delegate the tasks associated with these roles to other agents. Moreover, coherence allows agents to postpone the resolution of the inconsistencies among instances that cannot be fulfilled to a later iteration when the agent is able to fulfil some of the instances. Furthermore, several and repeated unsolved inconsistencies may be a sign of a problem with the normative system (e.g., norms may be impossible to fulfil), and agents may be able to propose a change to the normative system. Thus, unresolved inconsistencies can be considered as an opportunity to take into account more action possibilities such as delegating some tasks, reassigning roles, proposing a change to the normative system, etc.

It should be noted that improving agent capabilities for decision-making in resolving normative inconsistencies at run-time obviously comes at an additional computational cost. Specifically, coherence maximisation is an NP-complete problem [40]. In particular, the computational cost of computing coherence to resolve inconsistencies using an exhaustive algorithm that explores all possible solutions to the inconsistencies is given by $O(2^{|N_I|})$. The cost of resolving inconsistencies using a static order is given by $O(|N_I|)$ which is the

[23] We have considered alternative methods for calculating the conditional order (e.g., in the previous experiment about the activation and expiration beliefs we have also tried to calculate the conditional order as $\frac{\rho_s^F+\rho_c^F+\rho_{highTraffic(w)}-(1-\rho_{lowTraffic(w)})}{4}$), but these methods have also produced undesirable results.

cost of selecting the instance that has been created out of the most salient norm. Similarly, the cost of resolving an inconsistency using a conditional order is given by $O(|C|)$, where C is the set of conditions that determine under which circumstances each instance prevails. Using either the static order or conditional order to solve normative inconsistencies is clearly less computationally expensive than using the coherence approach, however, with these resolution strategies only one instance prevails, regardless of the number of inconsistent instances, while the coherence approach allows multiple instances to prevail. Moreover, coherence can be regarded as a suitable alternative to these resolution strategies because there already exist approximate resolution methods that can efficiently determine a set of elements to accept that have a satisfactory, if not maximal, degree of coherence [39]. Thus, agents with bounded rationality that may use a range of different algorithms, from exhaustive to incremental, for calculating coherence maximisation in order to resolve inconsistencies of normative instances where the number of instances determines the size of the problem and so the feasibility of using an optimal algorithm.

6 Experiments with Larger Sets of Instances

In the previous section we compared the performance of the coherence approach with approaches that determine which instance should prevail based on an order over the norms; since such approaches are only able to select the most preferred instance, the experiments in the previous section only considered two specific inconsistent instances. In these experiments a particular case of the inconsistency of instances (Definition 4) is considered. However, the support function for instances (Definition 16) calculates the incoherence degree in the same way for all cases of inconsistency between instances, which entails that these results can be generalised to other examples with two inconsistent instances.

A significant benefit of the coherence approach over approaches that only produce an order over the norms is that it is able to determine a subset of instances to comply with when faced with a set of instances in which there are several inconsistencies. In this section we analyse the behaviour of the coherence approach in such situations and consider the two different methods for calculating the coherence of partitions (sum of satisfied constraints and sum of satisfied constraints divided by the cardinality of the accepted set).

6.1 Experimental set up

We implemented a random scenario generator that takes as input $x \in \mathbb{N}$ and creates the elements of the coherence graph with x instances as follows.

- A set of atomic propositions P is created such that the size of P is randomly determined to be in the interval $[3, 10]$.
- A set of literals L is created such that for each atom in P we have a positive literal and a negative literal: i.e., $\forall p \in P : p \in L \wedge \neg p \in L$.
- A set of roles R is created such that the size of R is randomly determined to be in the interval $[1, 5]$.
- A set N of x norm expressions is created. For each norm expression $(n, \rho_s) \in N$:
  - the deontic modality of n is randomly selected from the set $\{\mathcal{O}, \mathcal{P}, \mathcal{F}\}$;
  - the norm condition of n is a single literal randomly selected from L;
  - the activation condition of n is a single literal randomly selected from L;

  - the expiration condition of n is a single literal randomly selected from L;
  - the target of n is a single role randomly selected from R;
  - ρ_s (the salience of n) is randomly assigned as a real value within the interval $[0, 1]$.
- For each norm $n = \langle D, C, T, A, E \rangle$ such that $(n, \rho_s) \in N$, an instance expression (i, ρ_c) is created such that $i = \langle D, C, T, A, E \rangle$ and ρ_c (the ease of compliance of i) is randomly assigned as a real value within the interval $[0, 1]$[24].
- For each role $r \in R$, a graded belief $belief(play(self, r), \rho)$ is created, such that the certainty of the belief ρ is randomly assigned as a real value within the interval $[0, 1]$.
- For each literal $l \in L$, a graded belief $belief(l, \rho)$ is created, such that the certainty of the belief ρ is randomly assigned as a real value within the interval $[0, 1]$.
- For each literal $l \in L$, a graded desire $desire(l, \rho)$ is created, such that the desirability of the desire ρ is randomly assigned as a real value within the interval $[0, 1]$.

Thus the random scenario generator produces a set of normative and cognitive elements that must be considered to resolve any inconsistencies in the instances. Note that the sets of propositions and literals are relatively small with respect to the number of instances that we consider; this ensures that there are likely to be inconsistencies between the instances and in fact we disregard any generated scenarios that do not contain at least one inconsistency. One might argue that it is not realistic to consider scenarios where there are many inconsistencies, since normative systems are typically designed with the aim of avoiding such situations; however, in open, non-deterministic, dynamic environments it is possible that such situations occur. Our coherence approach allows an agent to determine which subset of instances to comply with when faced with multiple inconsistencies, something that is not possible with existing approaches that depend on an order over the instances.

For each run of the experiment parametrised by x, a random scenario is generated with x instances as described above. The number inc of inconsistencies that exist in the scenario is calculated and then each of the *sum of satisfied constraints* and the *sum of satisfied constraints divided by the cardinality of the accepted set* methods is applied to calculate the coherence of all possible partitions, producing the partition that maximises coherence for each method. For each method, we calculate:

- the number x' of instances in the partition that maximises coherence;
- the number inc' of inconsistencies that exist in the partition that maximises coherence;
- the percentage of solved inconsistencies, i.e.,

$$\frac{inc - inc'}{inc} \times 100$$

- the percentage of instances that prevail, i.e.,

$$\frac{x'}{x} \times 100$$

For each $x \in \{2, \ldots, 19, 20\}$, we performed 1000 runs of the experiment.

6.2 Results

Our results show that, for each calculation method (*sum of satisfied constraints* and *sum of satisfied constraints divided by the cardinality of the accepted set*), the percentage of inconsistencies that are resolved is around 100%. This demonstrates that the coherence approach

[24] Note that since we are assuming that the norm, activation and expiration conditions are literals, the substitution for creating an instance from a norm is empty.

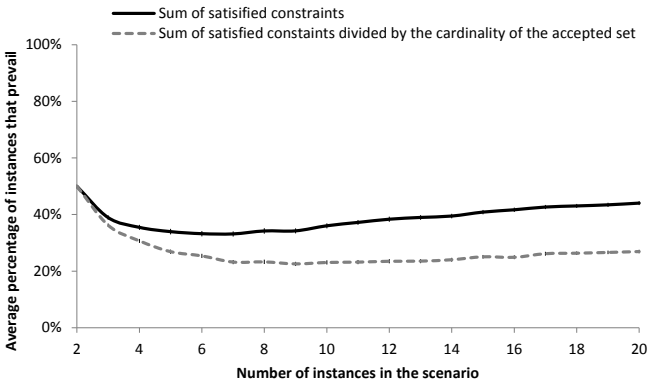


Fig. 11: Average percentage of instances that prevail out of 1000 runs over randomly generated scenarios with $n \in \{2, 3, \ldots, 19, 20\}$ instances, using both the *sum of satisfied constraints* and the *sum of satisfied constraints divided by the cardinality of the accepted set* methods to calculate coherence of partitions.

is well suited to dealing with situations in which there are multiple inconsistencies among a set of instances, which are those where it is not possible to determine which subset of instances should prevail by considering an order over the instances.

Figure 11 shows how the percentage of instances that prevail varies with the number of instances in the randomly generated scenario for each of the two methods for calculating coherence. The error bars represent the 95% confidence interval for the percentages obtained in the simulations. We see that the percentage of instances that prevail increases as the number of instances increases for the two calculation methods. In particular, this increase is more noticeable when coherence of partitions is calculated as the *sum of satisfied constraints*. Thus we see that when coherence of partitions is calculated as the *sum of satisfied constraints divided by the cardinality of the accepted set*, a smaller set of instances prevails than when coherence of partitions is calculated as the *sum of satisfied constraints*.

In general, we can conclude that the sum of satisfied constraints method is more robust in coping with a high number of instances, since the percentage of prevalent instances increases as more instances are considered. Thus, if an agent is endowed with enough resources (e.g., time) for complying with norms, it is more appropriate to apply the *sum of satisfied constraints* method for calculating coherence, since this allows a higher percentage of the instances to prevail. When the agent is not able to comply with several instances, it is more appropriate to calculate coherence of partitions as the *sum of satisfied constraints divided by the cardinality of the accepted set*, since this method accepts smaller sets of instances.

7 Related Work

In the existing literature, much work has also tackled the problem of resolving normative inconsistencies. For example, this problem has been addressed from a logical and formal perspective, e.g., the proposals contained in [29, 16] describe logical formalisms and axioms for representing and reasoning about conflicts in legal systems. Similarly, in [2, 24] mech-

anisms for checking the consistency of a normative system are proposed. In addition, this problem has been explored in the context of practical reasoning; i.e., there are proposals for the development of mechanisms to allow agents to select between inconsistent norms when they deliberate about the next action to be performed. Given that our proposal falls into this last category, this section reviews the most relevant proposals on inconsistency resolution in agents.

Firstly, we start this section by considering the work of Joseph et al. in [23] that has been taken as a reference for our proposal. Joseph et al.'s work is not focused on practical reasoning in agents under the influence of inconsistent norms. Instead, Joseph et al. propose a formalisation of deductive coherence for coherence-driven BDI agents that use coherence maximisation as a theory revision process. The example contained in [23] illustrates the use of coherence as a criterion for rejecting or accepting simple obligation norms. Thus, agents are responsible for extending the normative system, but they are not endowed with mechanisms for considering norms in practical reasoning; i.e., mechanisms for deciding what to do according to their goals and the norms. In contrast, our proposal addresses the problem of how normative agents under the influence of multiple possibly inconsistent instances make a decision about which instances to comply with. Specifically, we propose to resolve normative inconsistencies according to coherence. In this paper, we propose a formalisation of deductive coherence for resolving inconsistencies by means of a support function, as in Joseph et al.'s proposal. However, the main focus of our formalisation is to resolve normative inconsistencies and, as a consequence, the way in which we instantiate the coherence theory is different from the work of Joseph et al. For example, our normative definitions include normative elements such as activation and expiration conditions, norm target, salience and ease of compliance. All these elements play a key role in norm compliance decisions and should be considered when resolving normative inconsistencies. As a result, we propose new functions for calculating the coherence among nodes of the coherence graph that take into account the relationships among these normative elements and the cognitive elements. Our coherence graph only contains nodes that are relevant to resolve normative inconsistencies and only instances are rejected or accepted, which reduces noticeably the computational cost of the coherence maximisation process. We also propose different methods for calculating the coherence of partitions and resolving norm inconsistencies. Finally, our paper provides an experimental evaluation to analyse the performance of our solution for resolving inconsistencies across different examples and situations.

The problem of resolving inconsistencies among norms and other mental attitudes has been tackled by several authors. To name some examples, in [32] Modgil and Luck propose a framework for argumentation-based resolution of conflicts amongst desires and norms. In this framework, agents reason about arguments for and against compliance with norms. The topic addressed by our paper is slightly different. Here, we focus on resolving inconsistencies among instances. In our opinion, the solutions provided to this problem by the existing literature are too rigid to be used in dynamic and non-deterministic environments. For example, in [7] the authors assume that there is a norm hierarchy that determines the importance of norms and allows agents to resolve inconsistencies. Similarly, in [12] the authors assume that agents have a preference order among norms.

In [25], Kollingbaum describes NoA agents, which are agents governed by norms in their practical reasoning. Based on the characteristics of the NoA model, a classification and resolution strategy for inconsistencies among instances is presented. Specifically, the authors propose to resolve inconsistencies by: arbitrary decision, selecting the most recent instance, following the most restrictive instance, following the most general instance, following the most restrictive instance, or following the instance that has been created out of the most

salient norm. Some of these strategies are analysed in more detail in [26], and all of them imply that a static and predefined criterion is followed to resolve inconsistencies.

In [21] Gartner proposes a model of agency in which norms determine the agent behaviour. Gartner's work also takes into account the possibility that inconsistencies among norms may arise. As a solution to this problem, Gaertner proposes the use of an argumentation-based approach and a preference function that prioritises certain norms over others. Thus, one can consider preferences as describing the normative personality of an agent. However, these preferences are also specified by the agent programmer in a static way, entailing a limitation on the agents' capabilities for adapting to changing environments.

The BOID proposal [4] consists on an extension of the BDI architecture with an explicit notion of obligation. BOID agents violate norms only due to an inconsistency among obligations, desires or intentions. These inconsistencies are resolved by means of a static ordering function that is hard-wired in agents to determine which proposition prevails. Therefore, BOID agents always consider inconsistencies in the same manner; i.e., they cannot decide which norm prevails according to their current circumstances.

As demonstrated by our experiments in Section 5, even in simple cases where an agent has to make a decision between two inconsistent instances, using a static and predefined order based on the importance of norms does not produce satisfactory results. When agents belong to open and dynamic environments, static defined orders are unsuitable for resolving normative inconsistencies since the circumstances might change, making the predefined order obsolete [13]. Our experiments demonstrate that the dynamic circumstances in which norms are instantiated are an important factor when resolving normative inconsistencies, which cannot be captured by a static order or preference.

In a more recent proposal, Vasconcelos et al. [41] propose to avoid inconsistencies among instances by curtailing the scope of norms that may cause inconsistencies when they are instantiated. To determine which instance prevails in case of inconsistency, the authors use orders, or *policies*, that determine, given a pair of instances, which one is to be curtailed. Orders may include conditions that determine under which specific circumstances instances of a norm prevail. Thus, inconsistencies are resolved in a more elaborate way. As shown by Vasconcelos et al., classic forms of deontic conflict resolution, such as *lex posterior* (the most recent instance takes precedence) and *lex superior* (the instance imposed by the strongest power takes precedence) [28], can be represented as a conditional order. The solution proposed in our paper is somewhat similar to Vasconcelos et al.'s proposal, since both resolve inconsistencies by taking the agent's circumstances into account. However, as demonstrated by our experiments in Section 5 Vasconcelos et al.'s approach depends on conditional orders that are statically defined by the agent programmer off-line and may lose their validity at execution time, causing the conditional order approach to lead to undesirable results in certain situations.

All the aforementioned proposals determine that only one instance prevails (e.g., the instance that has been created out of the most salient norm) in case of inconsistency among several instances. In [34] Oren et al. propose the use of heuristics that have been defined inside argumentation theory [15] to solve normative inconsistencies among a set of instances in which several instances hold. Their work represents inconsistencies between instances as a graph in which the nodes are instances, and the arcs represent inconsistencies between instances. Basically, Oren et al. use a partial order among norms to determine which instances prevail in case of an inconsistency. When it is not possible to resolve all inconsistencies, then different heuristics are used to prune the graph and minimise the number of inconsistencies. Even though this work requires a predefined partial order among norms, it is one of the first proposals to deal with the resolution of inconsistencies among several instances.

However, it uses a very simple notion of norms as unconditional deontic propositions whose validity is taken for granted. As argued by Oren et al. in [34], this simple notion of norm does not allow relationships among norms to be specified. For example, it is not able to deal with groups of instances that are applicable under the same circumstances (e.g., a contract violation) and that may have to be considered together. Our proposal considers this type of relationship among norms by explicitly representing the circumstances under which norms are applicable. Thus, our coherence graph contains not only the instances and the inconsistency links, as in the proposal of Oren et al., but also norms and other cognitive elements that are relevant in resolving the inconsistencies.

8 Conclusion, Discussion and Future Work

Norms are used in multi-agent systems to restrict the potential excesses of agents' autonomous behaviours. The set of norms that are applicable to an agent changes during its execution and, as a consequence, it is possible that at some point the agent is affected by inconsistent instances. For this reason, normative agents require mechanisms that allow them to resolve these inconsistencies. Current proposals for resolving such inconsistencies assume that there is a predefined order among norms that determines which instance prevails in the case of inconsistency (i.e., the instance created out of the most salient norm prevails). A key drawback of these proposals is the fact that in the case of inconsistency just one instance prevails. Thus, it is not clear how agents use this order to select which instances prevail from a set of inconsistent instances. In addition, while a static order may be sufficient for multi-agent systems in which there is high compliance with specifications, in open multi-agent systems the performance of the system may be unpredictable, causing such a fixed order among norms to lose its validity.

In this paper we propose the first mechanism for resolving inconsistencies among sets of instances by computing a dynamic order that takes into account the cognitive and normative elements that are present in the agent knowledge base. Specifically, we propose a formalisation of deductive coherence that provides agents with a computational process for resolving inconsistencies in a dynamic and flexible way. To evaluate the performance of our proposal, we carried out several experiments that compare the resolution of inconsistencies between two instances by means of coherence and existing proposals. The results of these experiments demonstrate that coherence allows agents to resolve the inconsistencies satisfactorily by adapting the solution to different circumstances. Moreover, coherence allows agents to identify controversial situations in which the most coherent decision is to leave the inconsistency unsolved. For example, if an agent is unable to fulfil any of the inconsistent instances, coherence determines that the inconsistency remains unresolved and the resolution is postponed until the agent is able to comply with the inconsistent instances. Moreover, these controversial situations allow agents to detect situations in which it makes sense to reason about the delegation of tasks, the modification of the normative system, and so on.

We have demonstrated that our coherence approach is able to resolve inconsistencies among large sets of instances, showing that it resolves almost all inconsistencies. Moreover, the different ways in which the coherence of partitions can be calculated provides agents with different views of norm inconsistency. Specifically, we identified: a calculation method that provides solutions in which more instances are prevalent, and a calculation method that rejects more instances. The former is of special interest to open and highly dynamic multi-agent systems, since it is more robust with respect to the number of norms and instances. The latter may be of interest to agents that have limited capability to comply with instances. In

future work, we plan to test the usefulness of our proposal in real applications where agents are situated in highly changing environments. Moreover, we will perform experiments with several agents to assess the performance of coherence as a method to resolve normative inconsistencies in environments populated by heterogeneous agents that use different methods for this purpose.

References

1. C. E. Alchourrón and E. Bulygin. *Normative systems*. Springer-Verlag, 1971.
2. M. Aphale, T. Norman, and M. Sensoy. Goal-directed policy conflict detection and prioritisation. In H. Aldewereld and J. S. Sichman, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems VIII*, volume 7756 of *Lecture Notes in Computer Science*, pages 87–104. Springer Berlin Heidelberg, 2013.
3. P. Bourdieu. *Practical reason: On the theory of action*. Stanford University Press, 1998.
4. J. Broersen, M. Dastani, J. Hulstijn, Z. Huang, and L. van der Torre. The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In *Proceedings of the International Conference on Autonomous Agents*, pages 9–16. ACM, 2001.
5. M. Campenní, G. Andrighetto, F. Cecconi, and R. Conte. Normal= normative? the role of intelligent agents in norm innovation. *Mind & Society*, 8(2):153–172, 2009.
6. A. Casali, L. Godo, and C. Sierra. A graded BDI agent model to represent and reason about preferences. *Artificial Intelligence*, 175(7-8):1468–1478, 2011.
7. R. Conte and F. Dignum. From social monitoring to normative influence. *Journal of Artificial Societies and Social Simulation*, 4(2):7, 2001.
8. N. Criado, E. Argente, P. Noriega, and V. Botti. Human-inspired model for norm compliance decision making. *Information Sciences*, 245:218–239, 2013.
9. N. Criado, E. Argente, P. Noriega, and V. Botti. Manea: A distributed architecture for enforcing norms in open mas. *Engineering Applications of Artificial Intelligence*, 26(1):76–95, 2013.
10. N. Criado, E. Argente, P. Noriega, and V. Botti. Reasoning about constitutive norms in BDI agents. *Logic Journal of IGPL*, 22(1):66–93, 2013.
11. N. Criado, E. Argente, P. Noriega, and V. Botti. Reasoning about norms under uncertainty in dynamic environments. *International Journal of Approximate Reasoning*, 55(9):2049–2070, 2014.
12. F. Dignum, D. Kinny, and L. Sonenberg. From desires, obligations and norms to goals. *Cognitive Science Quarterly*, 2(3-4):407–430, 2002.
13. F. Dignum, D. Morley, E. A Sonenberg, and L. Cavedon. Towards socially sophisticated bdi agents. In *MultiAgent Systems, 2000. Proceedings. Fourth International Conference on*, pages 111–118. IEEE, 2000.
14. F.P.M. Dignum. Autonomous agents with norms. *Journal of Artificial Intelligence and Law*, 7(1):69–79, 1999.
15. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.
16. P. M. Dung and G. Sartor. The modular logic of private international law. *Artificial Intelligence and Law*, 19(2-3):233–261, 2011.
17. J. M. Epstein. Learning to be thoughtless: Social norms and individual computation. *Computational Economics*, 18(1):9–24, 2001.
18. F. Esteva and L. Godo. Monoidal t-norm based logic: towards a logic for left-continuous t-norms. *Fuzzy sets and systems*, 124(3):271–288, 2001.
19. M. Esteva, J. A. Rodrguez-Aguilar, C. Sierra, P. Garcia, and J. L. Arcos. On the formal specification of electronic institutions. In F. Dignum and C. Sierra, editors, *Agent Mediated Electronic Commerce*, volume 1991 of *Lecture Notes in Computer Science*, pages 126–147. Springer Berlin Heidelberg, 2001.
20. M. Fitting. *First-order logic and automated theorem proving*. Springer Verlag, 1996.
21. D. Gaertner. *Argumentation and Normative Reasoning*. PhD thesis, Imperial College, 2009.
22. S. Gottwald. Many-valued logic. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2014 edition, 2014.
23. S. Joseph, C. Sierra, M. Schorlemmer, and P. Dellunde. Deductive coherence and norm adoption. *Logic Journal of the IGPL*, 18:118–156, 2010.
24. T. C. King, V. Dignum, and M. B. van Riemsdijk. Re-checking normative system coherence. In T. Balke, F. Dignum, M. B. van Riemsdijk, and A. K. Chopra, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems IX*, Lecture Notes in Computer Science, pages 275–290. Springer International Publishing, 2014.

25. M. J. Kollingbaum. *Norm-governed Practical Reasoning Agents*. PhD thesis, University of Aberdeen, 2005.
26. M. J. Kollingbaum and T. J. Norman. Strategies for resolving norm conflict in practical reasoning. In *In Proceedings of the ECAI Workshop Coordination in Emergent Agent Societies*, pages 1–10, 2004.
27. M. J. Kollingbaum, T. J. Norman, A. Preece, and D. Sleeman. Norm conflicts and inconsistencies in virtual organisations. In P. Noriega, J. Vázquez-Salceda, G. Boella, O. Boissier, V. Dignum, N. Fornara, and E. Matson, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, volume 4386 of *Lecture Notes in Computer Science*, pages 245–258. Springer Berlin Heidelberg, 2007.
28. J. Leite, J. Alferes, and L. Pereira. Multi-dimensional dynamic knowledge representation. In T. Eiter, W. Faber, and M. Truszczyski, editors, *Logic Programming and Nonmotonic Reasoning*, volume 2173 of *Lecture Notes in Computer Science*, pages 365–378. Springer Berlin Heidelberg, 2001.
29. T. Li, T. Balke, M. De Vos, K. Satoh, and J. Padget. Detecting conflicts in legal systems. In Y. Motomura, A. Butler, and D. Bekki, editors, *New Frontiers in Artificial Intelligence*, volume 7856 of *Lecture Notes in Computer Science*, pages 174–189. Springer Berlin Heidelberg, 2013.
30. F. López y López, M. Luck, and M. d'Inverno. A normative framework for agent-based systems. *Computational & Mathematical Organization Theory*, 12(2):227–250, 2006.
31. F. Meneguzzi and M. Luck. Norm-based behaviour modification in BDI agents. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pages 177–184, 2009.
32. S. Modgil and M. Luck. Argumentation based resolution of conflicts between desires and normative goals. In I. Rahwan and P. Moraitis, editors, *Argumentation in Multi-Agent Systems*, volume 5384 of *Lecture Notes in Computer Science*, pages 19–36. Springer Berlin Heidelberg, 2009.
33. Y. Moses and M. Tennenholtz. Artificial social systems. *Computers and Artificial Intelligence*, 14(6), 1995.
34. N. Oren, M. Luck, S. Miles, and T. J. Norman. An argumentation inspired heuristic for resolving normative conflict. In *Proceedings of the Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems*, pages 41–56, 2008.
35. N. Oren, S. Panagiotidi, J. Vázquez-Salceda, S. Modgil, M. Luck, and S. Miles. Towards a formalisation of electronic contracting environments. *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, pages 156–171, 2009.
36. J.R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.
37. H. A. Simon. *Models of bounded rationality: Empirically grounded economic reason*, volume 3. MIT press, 1982.
38. M.P. Singh. An ontology for commitments in multiagent systems. *Journal of Artificial Intelligence and Law*, 7(1):97–113, 1999.
39. P. Thagard. *Coherence in thought and action*. The MIT Press, 2002.
40. P. Thagard and K. Verbeurgt. Coherence as constraint satisfaction. *Cognitive Science*, 22(1):1–24, 1998.
41. W. W. Vasconcelos, M. J. Kollingbaum, and T. J. Norman. Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 19(2):124–152, 2009.
42. D. Villatoro, G. Andrighetto, J. Sabater-Mir, and R. Conte. Dynamic sanctioning for robust and cost-efficient norm compliance. In *IJCAI*, volume 11, pages 414–419, 2011.
43. G.H. von Wright. *Norm and action: a logical enquiry*. Routledge & Kegan Paul, 1963.