

Commitments and interaction norms in organisations

Mehdi Dastani¹ · Leendert van der Torre² ·
Neil Yorke-Smith³

Published online: 27 November 2015
© The Author(s) 2015

Abstract In an organisational setting such as an online marketplace, an entity called the ‘organisation’ or ‘institution’ defines interaction protocols, monitors agent interaction, and intervenes to enforce the interaction protocols. The organisation might be a software system that thus regulates the marketplace, for example. In this article we abstract over application-specific protocols and consider commitment lifecycles as generic interaction protocols. We model interaction protocols by explicitly-represented norms, such that we can operationalise the enforcement of protocols by means of norm enforcement, and we can analyse the protocols by a logical analysis of the norms. We adopt insights and methods from commitment-based approaches to agent interaction as well as from norm-based approaches to agent behaviour governance. First, we show how to use explicitly-represented norms to model commitment dynamics (lifecycles). Second, we introduce an operational semantics to operationalise norm enforcement. Third, we show how to logically analyse interaction protocols by means of commitment dynamics and norm enforcement. The model, semantics, and analysis are illustrated by a running example from a vehicle insurance domain.

Keywords Commitments · Normative organisations · Operational semantics

The foundations of the ideas developed in this article were presented at the COIN workshop at AAMAS’12 [34] and briefly summarized at the AAMAS’12 conference [33].

✉ Neil Yorke-Smith
nysmith@aub.edu.lb

Mehdi Dastani
M.M.Dastani@uu.nl

Leendert van der Torre
leon.vandertorre@uni.lu

¹ Utrecht University, Utrecht, The Netherlands

² University of Luxembourg, Luxembourg, Luxembourg

³ American University of Beirut, Beirut, Lebanon

1 Introduction

A sizeable body of literature has developed theories about organisations, interaction protocols, commitments, and norms. For example, an organisational setting is manifest in the finance and accounting literature, where regulatory bodies monitor or audit firms, and may fine violations of regulations or statutes, and failures in corporate governance [26].

Andrighetto et al. [3] survey a body of work in multi-agent systems describing decision making and agent organisations using norms, interaction protocols and commitments. A central challenge in this literature is to investigate the relation between *commitment-based approaches* and *norm-based approaches*. This article contributes to this challenge by modelling *dynamics of commitments* (often explained as variety in commitment lifecycle) using an *explicit representation of norms*.

First, commitment-based approaches start from the insight that agents' interactions, including communication, create and process commitments according to what is called commitment lifecycle. Agents can thus interact, by virtue of which they may create commitments that may further be for example satisfied or violated. There exists variety in the details of the commitment lifecycle: for example, differing in the way commitments are created, in which agent can detach, delegate or cancel commitments, in when commitments are satisfied, or in how deadlines are enforced. In such approaches, a commitment lifecycle constitutes a generic interaction protocol describing sequences of actions, including speech acts, through which commitments are dynamically created and assigned to the interacting agents [43,62]. For example, consider the enactment of a vehicle insurance business protocol, where agents, including the repair garage, the assessor, the insurer and customer, interact and agree on the terms of an insurance [68]. This agreement and the further interactions between the agents dynamically create and assign commitments to the agents. We use this protocol as the running example in this article.

Second, norm-based approaches start from the general idea that norms regulate the agents' interaction by incentivizing norm-compliant behaviour and punishing non-compliant behaviour. In the literature on norms, it is generally assumed that norms are "explicitly represented", "agents can decide whether to follow the norms", "violation of norms may have consequences for the norm violating agents", and that norm-based systems specify "how and to which extent the agents can modify the norms" [3,15]. On the one hand, (artificial) agents need to explicitly represent norms, which are used in their deliberation and decision making process [1,17,58]. On the other hand, norms need to be explicitly represented outside the agents in an entity called the 'organisation' or 'institution', which monitors and coordinates agent interaction [10,12,18,30,32]. Moreover, norm-based systems need to be "organised by means of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfilment" [3,16,19,31,66]. In such approaches, a norm set constitutes an interaction protocol describing compliant and non-compliant sequences of actions.

The fact that a commitments lifecycle and a norm set can both be seen as constituting an interaction protocol suggests a fundamental relation between commitment lifecycle and norms. This relation, which forms the main focus of our article, can be explained by viewing a commitment cycle from a normative stance and by viewing it as a set of norms that govern the agents' interaction by means of commitments. Following this relation, we propose a formal framework where a commitment lifecycle is modelled as a set of explicitly represented interaction norms. This framework allows us to operationalize commitment lifecycles, and to logically analyse and compare them.

We address the following three research questions in this article:

1. **How to generalise commitment lifecycles by modelling them as a set of explicitly-represented interaction norms?** Agent interaction creates and affects the state of the commitments. This normative view allows us to formulate the conditions under which commitments are created and their states are affected by a set of norms. The norms, which are represented as counts-as rules, specify the dynamics of commitments. In addition, regulation policies, also represented as counts-as rules, are then introduced to cope with interactions that cause commitments to be violated.
2. **How to operationalise commitment lifecycles?** We present an operational semantics for norms. For our purposes, we assume two main processes: the *monitoring process* checks for conformance of agents' interactions to norms (that represent commitment dynamics), while the *enforcement process* intervenes to regulate the agents' interactions based on regulation policies.
3. **How to logically analyse and compare the commitment lifecycles?** Since commitment lifecycles are explicitly represented and operationalized as norms, we can adopt formal techniques to formally check the (dynamic) properties of commitment lifecycle and compare them. For example, we can analyse interference between distinct commitments, and redundancy of norms using techniques from deontic logic [44].

The scope of the article is on monitoring agents' interaction in an organisational setting. Other organisation attributes and processes, such as environmental interaction, or roles [5, 13], entities, and the relationships between them [6], are orthogonal to our purpose in this article. We therefore do not consider other issues discussed in the literature, such as explicit temporal references in the commitments, roles of the agents, teams and groups, organisational dynamics, procedural norms [14] and culture [36].

The layout of the article approximately follows the research questions. First, Sect. 2 specifies the organisational setting and defines agents' interaction through commitments, and illustrates this by a commitment lifecycle. Section 3 gives the formal framework of the operational semantics, showing how to formalise the organisation's monitoring and enforcement processes. Section 4 illustrates how to logically analyse interaction protocols and norms, by proving various properties about a commitment lifecycle. Section 5 demonstrates the generality of our methodology by considering alternatives of the commitment lifecycle. Section 6 situates our contribution in the literature. Section 7 discusses future work and Sect. 8 concludes the article.

2 Organisations, constitutive norms, and commitment lifecycles

In this article we consider an organisation as a coordination environment that supports and coordinates the interaction of a set of autonomous, rational decision-making entities called agents.¹ In order to support and coordinate agents' interactions, an organisation needs to monitor the agents' interactions and it possibly intervenes when they violate some norms such as their commitments towards each other. In the sequel, this general idea is explained in detail. We are indifferent to whether the organisation is either a *virtual* organisation or a *real* organisation [56].

¹ With its emphasis on autonomy, this notion of *agent* differs from that in principal-agent theory common in economics. At different levels of abstraction, a human being, a firm, a set of firms, and even a country can all be seen as agents.

In general, an organisation assumes a set of agents joined together in order to achieve their individual and joint objectives or interests. For example, agents may join in groups, coalitions, companies, institutions, or societies to realise some individual or joint objectives. Various detailed definitions of organisations have been developed in management science [64], sociology [69], and other social sciences. As usual in interdisciplinary research, there is no consensus on the best definition, which depends on the use of the concepts in a discipline. For computational purposes, organisations are typically modelled by, for example, concurrent coordinated systems, coordination mechanisms, collaborative systems, workflow systems, business processes, electronic institutions, or virtual environments [7,28,29,38,45,47,56,65–67].

In the context of this article, an organisation monitors and governs the interaction among agents by means of a set of what we call *interaction norms*. We also use the term *norm-based organisations* to emphasise the role of interaction norms in coordination. We conceive norm-based organisations as a constituent of normative multi-agent systems, i.e., “multi-agent system organised by means of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfillment” [3]. In other words, if multi-agent systems are comprised by a set of agents interacting with each other, then *norm-based organisations* provide a means to coordinate the activities of individual agents by means of norms [2,3,11,20,28,30].

2.1 Interaction norms based on commitments

Let us consider a setting where different agents interact by making offers to each other in a virtual organisational setting where the organisation monitors the offers, creates commitments and processes them based on the further behaviour of the agents, and generates responses when the agents behave in a non-compliant manner. For example, following a scenario in the literature [24], a car insurance agent (the Insurer) offers a customer agent (the vehicle Owner) that it will validate her car insurance if the premium is paid; it also offers a car repair garage (the Repairer) that it will approve the assessments and pay the repair costs of the cars for which the insurance have been validated; and it offers a car damage assessor (the Assessor) that it will pay for the assessment of car damages if he performs an assessment. Moreover, the assessor offers the repairer that he will assess car damages if the owner’s insurance is validated. Finally, the repairer offers the car owner that it will repair her damaged car if the car insurance is validated.

These offers create various commitments for the involved agents. The subsequent behaviours of the agents may either be according to their commitments—or they may violate the commitments. For example, the car owner may report an accident to her insurer and take the car to the garage. On certification that the insurance is valid, because the car owner has paid the insurance premium, the garage accepts the damaged car and contacts the assessor. Since the insurance is valid, the garage also commences the repair work on the vehicle. The assessor reports to the insurer, which approves the assessment of damage and pays the assessor for his work. The garage reports to the insurance company when it has completed the repair, and with the approval by the insurer of the assessment, the garage then tells the car owner that the car is ready. The insurer pays the garage.

In this scenario, the behaviours of the agents are perfectly aligned with their offers and commitments made and there is no need for the organisation to intervene because of violations of commitments. However, the agents may behave differently and not aligned with their commitments. For example, the garage might accept a damaged car that has no valid insurance. In such a case, for example, a warning should be sent to the garage or the collaboration

between the insurer and the garage should be reconsidered. We assume that these responses to commitment violations are generated by the (external) organisation that aims at supporting the coordination of activities of the involved agents. Note the nature of the organisation in the scenario. As described and courts of law notwithstanding, there is no real organisation invested with the power to monitor agent communication and actions and intervene if commitments are violated. Instead, one can imagine a virtual organisation entered into by all the agents, or an insurance marketplace that mediates the offers and their enactment.

The insurance claim scenario is described fully by Chopra et al. [24] based on documentation from the Financial Services Commission of Ontario (FSCO) on the claim process.² It shows a situation where a number of agents interact within an organisational setting, making various offers and thereby creating commitments that in turn govern their further activities. The important assumption of this article is that the creation of commitments as well as the evaluation of further behaviours (compliant or non-compliant) of the agents are subject to a set of *interaction norms*, and that the exact nature of these interaction norms can vary from one organisational setting to another. For example, an offer made by an agent can by itself create a commitment in one organisational setting, while in another organisational setting an offer can create a commitment only if both parties explicitly accept the commitments. We would like to emphasise that interaction norms are different from what we can call *domain-specific norms*, such as traffic or financial norms. In contrast to domain-specific norms, interaction norms create and process commitments among interacting agents based on their observed behaviours. The aim of this article is therefore to enable analysis and enactment of interaction norms in different organisational settings.

In our approach, the coordination between agents is realised exogenously at runtime by means of two continuous processes maintained by the virtual organisation: *monitoring* and *enforcement*. The organisation continuously strives to:

1. monitor the behaviour of interacting agents,
2. evaluate their behaviours based on interaction norms to generate and maintain commitments, and
3. take necessary measures (i.e., implement regulation policies) when commitments are not followed.

The organisation as we study it, being an exogenous process with authority over the agents, cannot *intervene* in the internal decision making of individual agents by, e.g., either disallowing them to perform actions or forcing them to perform specific actions. In other words, agents are autonomous in the sense that they choose their own actions. However, if the organisation with authority observes an agent not to behave in accordance to its commitments, then the organisation may intervene and implement a regulation policy, for example, by imposing a sanction on the agent. In such a case, we say that the agent ‘violates’ the norm, and that the violating behaviour potentially results in sanctions.

2.2 The commitment lifecycle

The dynamics of commitments depends not only on the organisational setting, as explained in the previous subsection, but it also depends on the structure of commitments. We consider commitments that are directional, conditional, and have deadlines. For example, the insurer agent can be committed to a car owner that if the premium has been paid by the car owner within two days, then the insurance will be validated within eight days. The dynamics of commitments is often explained by means of the so-called ‘commitment lifecycle’.

² www.fSCO.gov.on.ca/english/insurance/auto/afterautoaccidentENG.pdf.

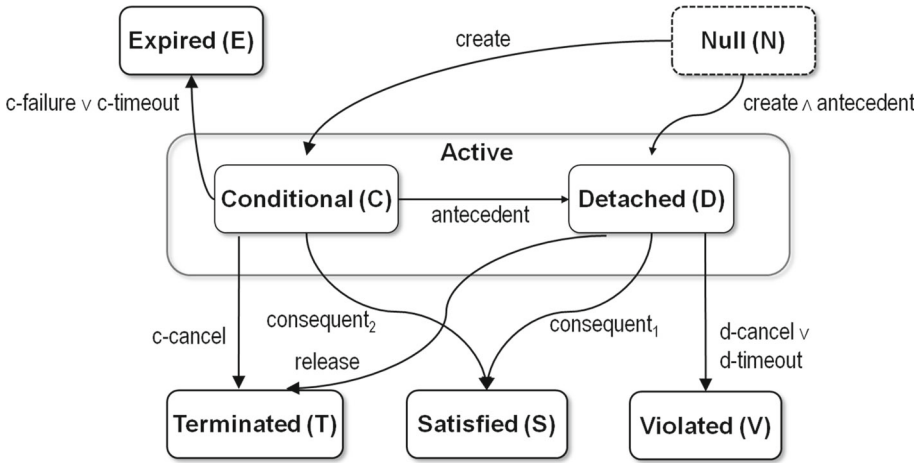


Fig. 1 State transitions of commitment lifecycle [65]

Figure 1 shows a commitment lifecycle adapted from Telang and Singh [65]. We omit here the suspension and delegation of commitments. In this figure, boxes indicate states of the commitments and arrows indicate transitions between commitment states. Consider an offer made by the insurer agent towards the car owner. This offer creates a conditional commitment, i.e., a commitment in, Conditional state. Should the antecedent of the commitment become true (i.e., premium is paid), the commitment moves to Detached state. Should the consequent of the commitment become true (i.e., insurance is validated), it moves to the Satisfied state. However, should the antecedent of the commitment not become true before its deadline (i.e., within two days), then the commitment moves to Expired state. Likewise if the consequent of a detached commitment does not become true by its deadline (i.e., within eight days), then the commitment move to Violated state. The commitment move also to Violated state if the committed agent (i.e., the insurer) cancels the commitment. Finally, the commitment moves to Terminated state if the committed agent cancels it before it moves to the Detached state, or if the second agent (i.e., the car owner) releases the committed agent (i.e., the insurer) once it has moved to the Detached state.

Let us anticipate some observations about commitment dynamics and lifecycles. First, we believe there is no unique canonical lifecycle for commitments, and in Sect. 5 we present various versions of commitment lifecycles. Second, we model a commitment lifecycle by a set of norms that govern the creation and processing of commitments, i.e., we model a commitment lifecycle by means of a set of *rules* representing the norms that govern the dynamics of commitments (see Sect. 3). Third, we assume that commitments can be created and their states can be changed due to either agents’ actions (communication or non-communication actions) or the state of the organisation. For example, the performance of an offer action by the insurance agent towards the customer agent creates a conditional commitment that the insurer will validate the insurance within eight days if the premium is paid by the customer with two days. This conditional commitment can be changed to a detached commitment if the premium is paid independent of whether the payment is done by the corresponding agent or in any other way as long as the payment is accomplished.

We represent interaction norms by counts-as rules. The representation of interaction norms by counts-as rules follows the idea of Searle [60] by distinguishing *brute* and *institutional*

facts [3, 59]. In our work, the commitments constitute the institutional facts. Using counts-as rules and considering commitments as institutional facts, interaction norms specify under which organisational circumstances commitments could be created and its states could be changed. In this way, counts-as rules that represent interaction norms govern the creation and processing of commitments. This allows us to express norms such as “an offer counts as a commitment”, or putting it in other words, “an offer creates a commitment”.

As stated in the introduction, our goal in this article is to provide a formal framework in which we can model, analyse and compare different commitment lifecycles. Our aim is neither to develop a detailed model of organisations nor to develop yet another computational model of normative multi-agent systems. Many such models have been proposed already, for example in the area of deontic logic in computer science [44], or in normative multi-agent systems workshops [3]. We believe that a commitment lifecycle provides a normative stance on agents’ interactions and can thus be used to exogenously coordinate agents’ interactions in organisational settings. Since there are various proposals on commitment cycles, we believe that the need is now for a generic framework which can accommodate different approaches to commitment cycles and offers the possibility to compare them. Hence we provide in the following section an operational semantics for a *generic organisation* that specifies how to monitor and regulate agents’ interactions without assuming a specific set of norms or regulation policies. In particular, the provided semantics assume a set of interaction norms that encode a specific commitment lifecycle and a set of regulation policies, and ensure that the interaction norms are enforced by means of regulation policies. The operational semantics is defined by transition rules that specify possible changes that an organisation can undergo at runtime. Since the interaction norms and regulation policies are assumed to be given and represented as rules, the transition rules are defined in terms of the application of rules.

3 Modelling commitment lifecycles by norms and regulation policies

We present in this section an operational semantics formalizing the two processes of the organisation, monitoring and enforcement, that manage agents’ interactions in an organisational setting.

As anticipated in the last section, our framework distinguishes brute facts from institutional facts. The brute facts denote either the facts about the physical environment of an organisation or the facts about the interaction between agents. The institutional facts denote commitments that are created between the agents. In the rest of this article, we assume a finite set of basic (brute and institutional) facts and a finite set of agents.

Definition 1 (*Environment facts*) Let Π_e be a finite set of propositional atoms denoting (basic) facts about the physical environment of an organisation. We assume $\top, \perp \in \Pi_e$. We define L_e to be the propositional language based on Π_e using logical connectives \neg and \vee .

In the rest of the article, we use variables ϕ, δ to stand for formulas from L_e .

A commitment lifecycle specifies the generation and dynamics of commitments based on agents’ interaction. Action Set 1 below consists of five representative actions that together specify the commitment lifecycle as presented in Fig. 1. As usual, we distinguish environment actions, i.e., actions that are performed by an agent in the physical environment of an organisation, from communication actions, i.e., actions that are performed by an agent to communicate with other agents. The set of communication actions does not have to be a complete set of speech acts, as in an agent communication language [43]: it just needs to

be sufficient for all state transitions in the commitment lifecycle, as is explained later in this section.

Action Set 1 Let \mathcal{A} be a finite set of agent names and x, y be variables to stand for agent names. The following is a set of communication action schemes. Each action scheme represents the set of all possible communication action instances. An action instance is an action scheme where each variable is instantiated with a specific value. In this case, an agent name variable is instantiated with a specific agent name and a variable for a formula from L_e is instantiated with a specific formula from L_e . For this set of communication action schemes and in the rest of this article, we abuse the notation of variables and use $\phi_x, \phi_y, \delta_x,$ and δ_y (instead of ϕ and δ) to stand for formulas from L_e . We use subscripts x and y to ease the reading of communication action schemes.

- offer($x, y, \phi_y, \phi_x, \delta_y, \delta_x$)— x offers y to make ϕ_x true (in the environment) by deadline δ_x if ϕ_y becomes true (in the environment) by deadline δ_y .
- tell(x, y, ϕ_x)— x tells y that x has made ϕ_x true (in the environment)
- cancel(x, y, ϕ_x)— x cancels its commitment to y to make ϕ_x true (in the environment)
- release(x, y, ϕ_y)— x releases the commitment of y to make ϕ_y true (in the environment)
- failure(x, y, ϕ_x)— x informs y that ϕ_x cannot be made true in the environment

Now the commitments are specified, let us look at one in detail.

Example 1 In our running example in the insurance scenario, an Insurer agent **I** can offer a Repairer agent **R** that it (the Insurer) will ‘approve the assessment’, denoted approve-assessment (or aa for short) within 4 days after the agent **R** ‘validates the insurance and reports the repair’, denoted insurance-validated \wedge report-repair (or iv \wedge rr for short). We use proposition d to denote the fact that the date of validation and report is a specific date and proposition $d+4$ to denote the fact that the current date is 4 days after the validation and report date. This offer action can be represented as offer(**I**, **R**, iv \wedge rr, aa, $d, d+4$).

We assume the existence of a set of action propositions that hold directly after performing the corresponding actions. An action proposition denotes the fact that an (environment or communication) action has been performed. For a given set of communication actions we assume a corresponding set of atomic communication action propositions that hold directly after performing the communication actions. Examples of atomic communication action propositions (corresponding to the communication action from Action Set 1) are the propositions offered($x, y, \phi_y, \phi_x, \delta_y, \delta_x$), told(x, y, ϕ_x), cancelled(x, y, ϕ_x), released(x, y, ϕ_y), and failed(x, y, ϕ_x). In the following, we call offered, told, cancelled, etc. the predicate names of the communication actions. Moreover, for a given environment action α we assume that the action proposition done(α) that holds directly after performing the environment action α .

Definition 2 (Action Propositions) Let L_e be the language of environment facts and ξ be a finite set of events consisting of a set of communication actions Com and a disjoint set of environment actions Act , i.e., $\xi = Com \cup Act$. Let also \mathcal{P} be the set of predicate names corresponding to the communication actions from Com . Π_{act} is the set of atomic (communication or environment) action propositions defined as follows:

$$\Pi_{act} = \{P(x, y, \phi_1, \dots, \phi_n) \mid P \in \mathcal{P} \ \& \ x, y \in \mathcal{A} \ \& \ \phi_1, \dots, \phi_n \in L_e\} \cup \{done(\alpha) \mid \alpha \in Act\}$$

In Definition 2 the assumption is implicitly made that communication (and more generally commitment) is always two party. A multicast message could be handled through many communicative actions, or the language could be extended with more action propositions.

We assume that atomic (communication or environment) action propositions are true in states following the corresponding (communication or environment) actions.

Example 2 After communication action $offer(l, R, iv \wedge rr, aa, d, d+4)$ the proposition $offered(l, R, iv \wedge rr, aa, d, d+4)$ will hold. We have the same assumption for the rest of communication actions. This communication action proposition denotes the fact that agent l has offered agent R to realise aa before deadline $d+4$ if $iv \wedge rr$ becomes true before deadline d . If the Insurer validates the insurance, then $done(iv)$ holds, and thus the antecedent of the offer is true.

In this article, institutional facts consist only of commitments. As introduced in the last section, a commitment expresses a social or organisational relationship between two agents. The view of commitments that we first represent follows that of Singh [62,65], in which commitments are unilateral, not requiring acceptance by the other party (but therefore not binding upon them). Later, in Sect. 5, we discuss an alternative view and show how our methodology can accommodate it also.

Further, as explained in Sect. 2.2, a commitment lifecycle assumes commitments to have states which change due to the commitment dynamics. We write commitments using the predicate C^s where the superscript s indicates that the commitment C is in state s . We assume a set C_{state} of commitment states (depending on the commitment lifecycle) and abbreviate the state names by the *commitment labels*. For example, for Fig. 1 we can have the following set of commitment states $C_{state} = \{n, e, c, d, t, s, vc, vt\}$. In this example set we distinguish violation due to cancellation (vc) and violation due to time-out (vt). We use predicate C without a superscript when the state of the commitment is not relevant. It is important to note that C is neither a second-order predicate nor a modal operator and that we do not aim at devising a logic to reason about the internal structure of commitments. We can now formally introduce commitments.

The following definition considers a what we call a commitment state to be an integral part of a commitment instance, in the sense that a commitment instance is always in a specific state. Hence, a commitment instance in a specific state, represented by a commitment proposition, is considered as an institutional fact. We do not consider a commitment state as an institutional fact.

Definition 3 (*Commitments*) Let L_e be the language of environment facts and C_{states} be a set of labels denoting possible commitment states. Π_C is the set of atomic commitment propositions defined as follows:

$$\Pi_C = \{C^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \mid s \in C_{states} \ \& \ x, y \in \mathcal{A} \ \& \ \phi_x, \phi_y, \delta_x, \delta_y \in L_e\}$$

As noted earlier, a commitment can be interpreted in various ways. For example, $C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ can be interpreted as agent x is committed to agent y to bring about ϕ_x before deadline δ_x if ϕ_y is brought about before deadline δ_y . This is Singh’s interpretation that we follow until Sect. 5. Another interpretation is that x is committed to y to bring about ϕ_x before δ_x if ϕ_y is brought about “by y ” before δ_y . The deadlines are represented by state propositions. A deadline is assumed to be reached in a state that satisfies it. For example, a deadline can be represented by the proposition ‘The date is Thursday, 21st May 2015’. This deadline is reached in a state where the date is Thursday, 21st May 2015.

Example 3 Returning to the insurance scenario, commitments between agents model the contracts and business protocols of the claim process as follows. Note there is an ‘implied’ commitment from insurer to customer: if the insurance has been validated and the car was damaged, then the car will have been repaired.

- C_1 (I to R): if insurance has been validated and the repair has been reported, then the insurer will have approved the assessment within 4 days and paid for the repair within 7 days
- C_2 (I to A): if the assessment has been done, then the insurer will have paid for the assessment within 5 days
- C_3 (A to R): if damages have been reported and the insurance has been validated, then a damage assessment will have been performed within 2 days
- C_4 (R to C): if the insurance has been validated and the car was damaged, then the car will have been repaired within 10 days
- C_5 (I to C): if the premium has been paid, then the insurance will have been validated within 8 days

Hence, formally, we have a set of commitments as follows. We model the conjunction in the consequent of C_1 by splitting C_1 to allow different deadlines for the two parts of the conjunction. Following Example 1, we use d_i to denote a specific date $i \in \mathbb{N}$ and $d_i + n$ to denote n days after the date d_i .

- $C_{1a} = C(I, R, \text{insurance-validated} \wedge \text{repair-reported}, \text{assessment-approved}, d_1, d_1 + 4)$
- $C_{1b} = C(I, R, \text{insurance-validated} \wedge \text{repair-reported}, \text{payment-done}, d_1, d_1 + 7)$
- $C_2 = C(I, A, \text{assessment-done}, \text{assessment-paid}, d_2, d_2 + 5)$
- $C_3 = C(A, R, \text{damages-reported} \wedge \text{insurance-validated}, \text{assessment-done}, d_3, d_3 + 2)$
- $C_4 = C(R, C, \text{insurance-validated} \wedge \text{car-damaged}, \text{car-repaired}, d_4, d_4 + 10)$
- $C_5 = C(I, C, \text{premium-paid}, \text{insurance-validated}, d_5, d_5 + 8)$

Example 4 In our running example, consider the first commitment $C_{1a} = C(I, R, iv \wedge rr, aa, d_1, d_1 + 4)$ denoting the fact that the insurer is committed to approve the assessment within 4 days after the repairer validates the insurance and reports the repair. In other words, agent $I \in \mathcal{A}$ (as debtor) tells agent $R \in \mathcal{A}$ (as creditor) that if proposition $iv \wedge rr$ (the antecedent) is brought about by deadline $d_1 \in L_b$ then I will bring about $aa \in L_b$ (the consequent) by deadline $d_1 + 4 \in L_b$.

We can now define the language of brute and institutional facts.

Definition 4 (*Brute/institutional facts*) Let Π_e , Π_{act} , and Π_c be the sets of atomic environment, action, and commitment propositions, respectively. $\Pi_b = \Pi_e \cup \Pi_{act}$ is the set of atomic propositions denoting atomic brute facts and $\Pi_i = \Pi_c$ is the set of atomic propositions denoting atomic institutional facts. Finally, $\Pi = \Pi_b \cup \Pi_i$ is the set of propositional atoms. We define L_b to be the propositional language based on Π_b using logical connectives \neg and \vee , and L to be the propositional language based on Π using logical connectives \neg and \vee .

In the rest of the article, we use variables φ, ψ to stand for formulae from L .

3.1 Norms and regulation policies: syntax

In our model of an exogenous organisation, we assume that individual agents interact with each other and with their shared physical environment, the agents’ interactions are primarily responsible for the creation and dynamics of commitments, and that the organisation

monitors the agents' interactions and manages the creation and dynamics of the commitments. In the rest of this section, we propose to model commitment lifecycles by means of norms and regulation policies. We present a generic but expressive representation for norms and regulation policies to model a broad class of commitment lifecycles. We then specify the exogenous processing of commitments by providing the operational semantics that effectuate the agents' interactions in the context of norms and policies that are designed to model a specific commitment lifecycle. The agents' (inter)actions are assumed to be observed/received by the organisation as external events. The evaluation of agents' (inter)actions as well as the processing of commitments by the organisation are realised based on the norms and regulation policies that model a commitment lifecycle.

3.1.1 Interaction norms

A commitment lifecycle can be considered as constituting a set of norms that govern the agents' (inter)actions. In other words, a commitment lifecycle can be seen as normatively specifying the conditions under which agents' interactions create commitments and process their dynamics. It is these commitment based norms that are used by an organisation to evaluate the agents' interactions to manage commitments. Thus, norms specify how agents' interactions cause the creation and dynamics of commitments.

Following Searle's tradition [60], we represent norms by counts-as rules. Searle's original formulation of the counts-as construct is of the form " φ counts as ψ in the context c ". In particular, we use ψ to represent a commitment (with a particular state), φ to represent the condition under which commitment ψ should be created or modified, and c to represent the context. In our case, a context can be, for example, a specific commitment lifecycle that constitutes the norms, together with the ontology that is used to specify the norms for the specific application domain such as the insurance scenario. In general, we assume that the contexts can be defined by the system designer. In the following, we use \mathcal{C} to be the set of all contexts.

Below the subscript n in " \Rightarrow_n " indicates that the rule represents a norm. This is due to the use of a similar rule for representing regulation policy later in this section. Thus, the use of subscript n for norms and later on r for regulation policies distinguishes norms and regulation policies.

Definition 5 (Norm) Let $\mathcal{C} = \{c_1, \dots, c_m\} \cup \{\top\}$ be a finite set (of contexts). A *norm* is an expression of the form " $\varphi \Rightarrow_n \psi$ in c ", where $\varphi \in L$, $\psi \in \Pi_i$, and $c \in \mathcal{C}$. For brevity we write " $\varphi \Rightarrow_n \psi$ " for " $\varphi \Rightarrow_n \psi$ in \top ". We use \mathcal{N} to denote the set of all norms.

Since ψ can only be an institutional fact, the norm as defined here can be considered as a constitutive norm specifying the institutional interpretation of an organisational state that satisfies φ and the context c . It should also be noted that when φ and ψ involve one and the same commitment with different state annotations, the norm specifies the state change of the commitment.

Example 5 Figure 2 is an example set of norms that models the lifecycle of commitments as illustrated in Fig. 1. We use here the set of commitment state labels $S = \{c, d, s, vc, t, e\}$ and the set of communication actions as presented in Action Set 1. These norms are explained below.

1. Performing action " x offers to y that x realizes ϕ_x before δ_x if y realizes ϕ_y before δ_y " counts as creation of a conditional commitment. The application of this rule,

- 1 $offered(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \Rightarrow_n C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_1
- 2 $told(y, x, \phi_y) \wedge C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg \delta_y \wedge \phi_y \Rightarrow_n C^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_1
- 3 $told(x, y, \phi_x) \wedge C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg \delta_y \wedge \phi_x \Rightarrow_n C^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_1
- 4 $told(x, y, \phi_x) \wedge C^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg \delta_x \wedge \phi_x \Rightarrow_n C^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_1
- 5 $cancelled(x, y, \phi_x) \wedge C^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg \delta_x \wedge \neg \phi_x \Rightarrow_n C^{vc}(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_1
- 6 $failed(y, x, \phi_y) \wedge C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg \delta_y \wedge \neg \phi_y \Rightarrow_n C^e(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_1
- 7 $cancelled(x, y, \phi_x) \wedge C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg \delta_y \wedge \neg \phi_y \Rightarrow_n C^t(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_1
- 8 $released(y, x, \phi_y) \wedge C^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg \delta_x \wedge \phi_y \Rightarrow_n C^t(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_1

Fig. 2 Example context 1 (c_1). Counts-as rules specify the lifecycle of commitments based on communication actions

which we refer to as rule 1, by the organisation will add conditional commitment $C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, considered as an institutional fact, to the set of institutional facts. Superscript c denotes conditional state of commitment; similar convention is used for other commitment states.

2. Performing action “y tells x that ϕ_y is realized” when δ_y is still not passed (does not hold) counts as detaching the conditional commitment. The application of this rule 2 will lead to the removal of the conditional commitment from institutional facts and adds a corresponding detached commitment to it. Detaching a commitment based on telling assumes that y is a trusted agent, i.e., its utterances are according to its beliefs. An organisation may develop a list of trusted agents.
3. Performing action “x tells y that ϕ_x is realized” when δ_y is still not passed counts as satisfying the conditional commitment. The application of this rule 3 removes the conditional commitment from institutional facts and adds a corresponding satisfied commitment to it.
4. Performing action “x tells y that ϕ_x is realized” when δ_x is still not passed counts as satisfying the detached commitment. The application of this rule 4 removes the detached commitment from institutional facts and adds a corresponding satisfied commitment to it.
5. Performing action “x cancels to realizes ϕ_x ” when δ_x is still not passed counts as the violation of the detached commitment. The application of this rule 5 removes the detached commitment and adds a corresponding violated commitment.
6. Performing action “y fails to realize ϕ_y ” when δ_y is still not passed counts as expiration of the conditional commitment. The application of this rule 6 removes the conditional commitment and adds a corresponding expired commitment.
7. Performing action “x cancels to realize ϕ_x ” when δ_y is still not passed counts as termination of the conditional commitment. The application of this rule 7 removes the conditional commitment and adds a corresponding terminated commitment to it.
8. Performing action “y releases a detached commitment after ϕ_y has been satisfied” when δ_x is still not passed counts as termination of the conditional commitment. The application of this rule 8 removes the detached commitment from institutional facts and adds a corresponding terminated commitment to it.

Note that Fig. 2 does not contain norms that are based on environment actions, i.e., actions that when performed change the factual/physical state of the environment. Such norms, which depend on the application domain at hand, can be represented by counts-as rules as well. Figure 3 gives some domain-dependent norms represented as counts-as rules. For example, the first norm indicates that a conditional commitment with ϕ_y as its condition counts-as a

- 1 $done(\alpha(y)) \wedge C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg\delta_y \wedge \phi_y \Rightarrow_n C^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_2
- 2 $done(\alpha(x)) \wedge C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg\delta_y \wedge \phi_x \Rightarrow_n C^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_2
- 3 $done(\alpha(x)) \wedge C^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg\delta_x \wedge \phi_x \Rightarrow_n C^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_2

Fig. 3 Example context 2 (c_2). Counts-as rules specify the lifecycle of commitments based on non-communication actions. $\alpha(x)$ denotes an action by agent x

- 1 $C^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \delta_x \Rightarrow_n C^{vt}(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_3
- 2 $C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \delta_y \Rightarrow_n C^e(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_3

Fig. 4 Example context 3 (c_3). Counts-as rules specify the lifecycle of commitments based on environment events. Commitments have to be fulfilled strictly before the deadline is true

detached commitment whenever condition ϕ_y is satisfied by some domain action α performed by the corresponding agent y before deadline proposition δ_y .

We allow brute facts to change as a consequence of the internal mechanism of the environment, e.g., the state of a clock changes automatically. Such changes may have direct impacts on commitments in the sense that the state of the existing commitments may change. For example, a commitment may change its state from detached to violated if the environment make a transition (time elapses) through which the deadline (a brute fact) becomes true. Note that such commitment dynamics do not depend on agents' actions, but they are rather based on state transitions due to the internal mechanism of the environment. Such dynamics of commitments can also be captured through norms represented by counts-as rules.

Figure 4 lists some examples of norms that govern the dynamics of commitments due to the internal mechanism of the environment. Note that the antecedent of these counts-as rules do not require the performance of any agent action.

- The elapse of deadline δ_x counts as the violation of a detached commitment. The application of this rule 1 in Fig. 4 removes the detached commitment from institutional facts and adds a corresponding violated commitment to it.
- The elapse of deadline δ_y counts as expiration of a conditional commitment. The application of this rule 2 in Fig. 4 removes the conditional commitment from institutional facts and adds a corresponding expired commitment to it.

3.1.2 Interaction regulation

Norms specify how agents' interactions cause the creation and dynamics of commitments. In addition to norms, we introduce the concept of regulation policy to allow an organisation to intervene in agents' interactions or to enforce commitment dynamics. The idea of regulation policy is thus to allow the management of commitments to be independent, or indirectly dependent, on the agents' interactions. For example, a regulation policy in the insurance scenario may intervene and send a warning to a garage (or even to terminate its collaboration with insurance companies) that has accepted damaged cars without valid insurances. Another regulation policy may be to invalidate an existing insurance because of an unpaid bill, which in turn may cause a commitment to be violated unless the payment is fulfilled immediately. In this sense, a regulation policy can be used to model application-dependent issues that are relevant for the management of commitments. We stress that although most existing commitment lifecycles can be modelled by a set of norms, the introduction of regulation policy enriches and extends the class of commitment lifecycles with application dependent issues that are relevant for the creation and dynamics of commitments. Following the representation

of norms, regulation policies can be represented by means of rules as well. As noted before, we use subscript r to distinguish rules representing regulation policies from rules representing norms.

Definition 6 (*Regulation policy*) Let $\overline{\Pi_e}$ be the set of negated atomic facts about the physical environment, i.e., $\overline{\Pi_e} = \{\neg p \mid p \in \Pi_e\}$. A *regulation policy* is an expression of the form “ $\varphi_b \wedge \varphi_i \Rightarrow_r \psi$ in c ”, where $\varphi_b \in L_b$, $\varphi_i = p_1 \wedge \dots \wedge p_n$ for $p_i \in \Pi_i$ and $1 \leq i \leq n$, and $\psi \in \Pi_e \cup \overline{\Pi_e}$, and $c \in \mathcal{C}$. For brevity we write “ $\varphi \Rightarrow_r \psi$ ” for “ $\varphi \Rightarrow_r \psi$ in \top ”. We use \mathcal{R} to denote the set of all regulation policies.

The antecedent of a regulation policy is a conjunction of φ_b , which denotes a brute fact, and φ_i , which denotes a an institutional fact. The institutional fact is in turn represented by a conjunction of institutional atoms. The reason for keeping brute and institutional facts separately, and taking the conjunction of institutional atoms instead of arbitrary institutional formula, is to allow removing institutional facts after the policy being applied (see Definition 11 for more details). As we explain later, removing institutional facts may influence the application of other regulation policies that contain some of the removed institutional atoms in their antecedent. Therefore, we emphasize that the regulation policies should be designed cautiously by taking the operational semantics and the applications of regulation policies into account.

This definition does not allow a regulation policy to create new commitments in order to respond to the existing institutional facts. For example, a regulation policy can specify a response that warns a garage which violates its commitment to a car owner to repair her car with valid insurance, but a regulation cannot create a new commitment that *obliges* the garage to still repair the car (recall the discussion at the end of Sect. 2.1). However, a new commitment can be created as the consequence of some institutional facts by means of norms (often seen as ‘contrary-to-duty’ norms). Thus, we consider regulation policy as changing the brute state (and not institutional state) of the organisation; the institutional state of an organisation can be changed merely by means of norms.

Example 6 Here we give some examples of regulation policies for the insurance scenario. We use c_{10} to denote the context of the insurance scenario, which is exactly the combined context of c_1, c_2 and c_3 . Recall that we do not perform calculation with deadlines, but assume some kind of calendar logic [54].

1. The Insurer’s commitments towards the Assessor and Repairer come with penalty policies, by which the Insurer have to pay the Assessor a penalty if the Insurer is late in approving a valid assessment and pay the Repairer a penalty if the Insurer is late in paying for a completed repair. These penalties are modelled by the following two regulation policies:

$$\begin{aligned}
 C_{1a}^{vt} &\Rightarrow_r \text{pay}(I, A, p_{1a}) \text{ in } c_{10} \\
 C_{1b}^{vt} &\Rightarrow_r \text{pay}(I, R, p_{1b}) \text{ in } c_{10}
 \end{aligned}$$

where p_{1a} and p_{1b} are the amounts of the respective penalties. Note that we use C_{1a}^{vt} to abbreviate commitment $C_{1a}^{vt}(I, R, iv \wedge rr, aa, d_1, d_1 + 4)$. We also use C_{1b}^{vt} as an abbreviation of the commitment with the same arguments as in C_{1a}^{vt} . This notational choice is used for the following regulation rules as well.

There are also penalties if the Insurer fails to approve the assessment or pay for the repair, respectively, because it cancels the commitment:

$$\begin{aligned} C_{1a}^{vc} &\Rightarrow_r \text{pay}(I, A, p'_{1a}) \text{ in } c_{10} \\ C_{1b}^{vc} &\Rightarrow_r \text{pay}(I, R, p'_{1b}) \text{ in } c_{10} \end{aligned}$$

where p'_{1a} and p'_{1b} are the amounts of the respective penalties.

2. The Insurer's commitment towards the Assessor comes also with a penalty policy, by which the Insurer will pay the Assessor a penalty if the Insurer is late in paying for a completed assessment. This penalty is modelled by the following regulation policy:

$$C_2^{vf} \Rightarrow_r \text{pay}(I, A, p_2) \text{ in } c_{10}$$

where p_2 is the amount of the penalty. There is a further regulation policy that penalizes the Insurer for cancelling the commitment, similar to C_{1a} and C_{1b} .

3. We suppose a penalty policy in the insurance scenario, whereby the Insurer guarantees that properly-reported eligible car damage will be repaired within 10 days. This applies to commitment C_4 . Note that it is the Insurer who makes this guarantee to the Customer, not the Repairer. One could model the guarantee by an additional commitment from Insurer to Customer (not shown); the deadline on the consequent in C_4 derives from this. As we have it the guarantee is modelled in C_4 and the following regulation policy:

$$C_4^{vf} \Rightarrow_r \text{pay}(I, C, \text{€ } 200) \text{ in } c_{10}$$

which is applied for any delay beyond the deadline. This rule models the penalty clause upon the Insurer which states that the Insurer will compensate the Customer for delayed repairs by the amount of 200 euros. There is a further regulation policy that similarly penalizes the Insurer for cancelling the commitment.

4. We suppose that the Insurer is part of the national association of insurers. The best practice guidelines of the association state that an insurer should validate a customer's insurance within eight days, provided the customer has pay the premium. This is reflected in C_5 . The association will send a reprimand to the Insurer if it violates its guidelines. The following regulation policy models this:

$$C_5^{vf} \Rightarrow_r \text{reprimand}(\text{Association}, \text{Insurer}) \text{ in } c_{10}$$

As with the other commitments, there is a similar regulation policy if the Insurer cancels the commitment.

Note that in our operational semantics, violation states are deleted once the regulation policies have been applied. Hence with the norms we consider in this article, one cannot model policies according to which violations are sanctioned repeatedly or continually until the commitment is satisfied. For example, we cannot model a penalty such as reducing the payment for the assessment by 20% for each day it is late (up to a maximum of 5 days delay, whereupon no payment is due). However, our framework can model repeated regulation, if a different set of norms is adopted in the operational semantics.

A special use of regulation policy is to *regiment* norms, i.e., make it impossible to violate the norms [52]. This regimentation is realized by preventing the creation of institutional facts related to the norm. For example, one can regiment the norm that a commitment cannot be cancelled by preventing the creation of the institutional fact C^{vc} . In our framework, a norm can be regimented by a regulation policy, represented by a rule, where the institutional fact that is generated by the norm is the antecedent of the rule and the brute fact \perp is the

consequent of the rule. Our example of regimenting the norm not to cancel a commitment can be represented by the following regulation policy rule:

$$C_{1a}^{vc} \Rightarrow_r \perp$$

As discussed below in Sect. 3.2.1 on operational semantics, the application of a regulation policy that is designed to regiment a norm will not cause any inconsistency in our framework. As explained there, the application of rules with \perp in the consequent does not add \perp to the organisation state to cause inconsistency, but it prevents the organisation to make a transition. In other words, the operational semantics assure that the actions that trigger such a policy to be applied will have no effect and will therefore not cause any inconsistency.

An organisation can now be specified in terms of brute facts, a specific organisational context, a set of norms, and a set of regulation policies.

Definition 7 (*Organisation specification*) Let \mathcal{N} be the set of all norms and \mathcal{R} be the set of all regulation policies, as defined in Definitions 5 and 6, respectively. An *organisation specification* is a tuple (F, \mathbf{c}, N, R) , where $F \subseteq \Pi_b$ is a finite set of brute facts, $\mathbf{c} \subseteq \mathcal{C}$ is a finite set of contexts, $N \subset \mathcal{N}$ is a finite set of norms, and $R \subset \mathcal{R}$ is a finite set of regulation policies.

Example 7 The organisation in the insurance claims scenario can be specified by $(\{\mathfrak{d}_0\}, \mathbf{c}, N, R)$, where $\mathbf{c} = \{c_{10}\}$, N consists of the norms listed in Example 5 and R consists of the policies listed in Example 6. The set of brute facts is initially assumed to contain only the state proposition \mathfrak{d}_0 denoting the initial date of the organisation is a certain date.

3.2 Norms and regulation policies: operational semantics

We build up the operational semantics for an organisation through several steps. We start by defining an organisation state and derive the initial state of an organisation from its specification. The execution of an organisation—during which the organisation monitors events, evaluates them, and intervenes based on regulation policies—changes the state of the organisation and results in a sequence of organisation states. In the following, we use ξ as the set of events including agents' actions performed in a shared environment, agents' communication, and changes of the state of the shared environment caused by its internal mechanism (e.g., the passing of the time). Such changes are important as they may bring some commitments to a violation state. For example, the elapse of time can cause a deadline to be reached which consequently may make a commitment be violated. In order to define the executions of an organisation, we present some transition rules that specify possible execution steps.

Definition 8 (*Organisation state*) Let $\sigma_b \subseteq \Pi_b$ be a finite set of brute facts, let $\sigma_i \subseteq \Pi_i$ be a finite set of institutional facts, $\mathbf{c} \subseteq \mathcal{C}$ a finite set of contexts, $N \subset \mathcal{N}$ a finite set of norms, and $R \subset \mathcal{R}$ a finite set of regulation policies. A *state of an organisation* is a tuple $(\sigma_b, \sigma_i, \mathbf{c}, N, R)$.

The initial state of an organisation is determined by its specification (see Definition 7). We assume that institutional facts are generated at runtime such that an organisation initially has no institutional facts. Finally, we assume that the context, norms, and regulation policies are given and do not change at runtime. The latter assumption can be relaxed by including additional meta-policies that indicate how the context, norms, and regulations can change at runtime. We leave aside this aspect, and the related issue of dynamic role assignment [32], as our aim is to focus on normative and regulative aspects of organisations.

Definition 9 (*Initial organisation state*) Let (F, \mathbf{c}, N, R) be the specification of an organisation. The initial state of the organisation is $(F, \emptyset, \mathbf{c}, N, R)$, i.e., the initial brute facts is given by the organisation specification and the initial institutional facts is the empty set.

For brevity, we omit N, R , and \mathbf{c} from the organisation states and represent an organisation state as (σ_b, σ_i) . However, we use N, R , and \mathbf{c} in the transition rules to determine the next organisation state based on the current state, and assume they are static ingredients of the configurations.

Example 8 Consider the insurance organisation as specified in Example 7. The initial state of the insurance organisation is $(\{\text{d}_0\}, \emptyset, \mathbf{c}, N, R)$. Assuming that \mathbf{c}, N , and R do not change during the execution of the organisation, we omit them from the organisation states in the following examples.

3.2.1 Transition rules

The behaviours of a set of agents that interact with each other (communication) and with a shared environment, as well as the behaviour of the shared environment caused by its internal mechanism, are processed in the context of an organisation, which is specified by a set of norms and a set of regulation policies. In particular, these behaviours are processed by applying norms and regulation policies. The application of norms and regulation policies constitutes a possible execution step of an organisation. In order to capture such an execution step by means of a transition rule, we first define an application function that determines applicable rules.

Definition 10 (*Applicable rules, antecedent and consequent of rules*) Let λ be a finite set of either norms or regulation policies (i.e., either $\lambda \subset \mathcal{N}$ or $\lambda \subset \mathcal{R}$), $R \subset \mathcal{R}$, Π the set of propositional atoms, Π_i be the set of propositional atoms denoting institutional facts, and \mathbf{c} a set of contexts. The function $\text{App} : 2^\Pi \times 2^\lambda \times 2^{\mathcal{C}} \rightarrow 2^\lambda$ selects a subset of rules from λ that are applicable in states specified by $X \in \Pi$ and contexts \mathbf{c} . Moreover, we specify the functions $\text{cond} : 2^{\mathcal{R}} \rightarrow 2^{\Pi_i \cup \overline{\Pi_i}}$ and $\text{cons} : 2^\lambda \rightarrow 2^{\Pi \cup \overline{\Pi}}$ to collect propositional atoms involved in the antecedent and consequent of a set of (applicable) rules, and define them as follows:

- $\text{cons}(\lambda) = \{\psi \mid (\phi \Rightarrow \psi \text{ in } c) \in \lambda\}$, and
- $\text{cond}(R) = \{p_1, \dots, p_m \mid (\varphi_b \wedge p_1 \wedge \dots \wedge p_m \Rightarrow_r \psi \text{ in } c) \in R, \varphi_b \in L_b, p_i \in \Pi_i\}$,

Note that the function cond is only applicable to regulation policies, while the function cons is applicable to both norms as well as regulation policies.

There are various ways to define the application function. Examples 9 and 10 illustrate two possibilities for defining the App function.

Example 9 Let $X \subseteq \Pi$ be a finite set of propositional atoms, λ a finite set of rules (either $\lambda \subset \mathcal{N}$ or $\lambda \subset \mathcal{R}$), s_X the state that is specified by X , \mathbf{c} a set of contexts. The following example of the App function does not involve iteration of rules applications.

$$\text{App}(X, \lambda, \mathbf{c}) = \{(\varphi \Rightarrow \psi \text{ in } c) \in \lambda \mid s_X \models \varphi \ \& \ c \in \mathbf{c}\}$$

There are various ways to determine s_X . One possibility is by applying the closed-world assumption, i.e., $\forall p \in \Pi : s_X \models p$ if $p \in X$ and $s_X \models \neg p$ if $p \notin X$.

A disadvantage of this definition of **App** is that the applicability of one rule does not trigger other rules to become applicable, i.e., rule application is assumed not to iterate. The iteration of rule application may be of particular importance for computing the consequences of the so-called interlocking norms (e.g., contrary-to-duty norms). These norms are often represented by rules such that the consequent of one rule is the antecedent of other rules. An advantage of the non-iterative definition is, however, its low computational load. Following this definition, the computation of norm consequences is linear in the size of the norm set.

The following example obtains iteration by taking the *closure* of applicable rules for a set of propositional atoms and a context. The closure of applicable rules from a set of rules λ , a set of propositional atoms X and a set of contexts \mathbf{c} is similar to the **App** function as defined above to compute applicable rules in one round. The closure ensures that the outcome of one round is used to compute applicable rules in the next round.

Example 10 Let $\text{cl}_X^{\lambda, \mathbf{c}}(Y) = \{(\varphi \Rightarrow \psi \text{ in } c) \in \lambda \mid s_{X \cup \text{cons}(Y)} \models \varphi \ \& \ c \in \mathbf{c}\}$, i.e., given a set of applicable rules $Y \subseteq \lambda$ the next set of applicable rules from λ are determined by also taking the consequent of the rules in Y . The iterative version of the rule application function, also represented by the function **App**(X, λ, \mathbf{c}), can now be defined as the least fixed-point of $\text{cl}_X^{\lambda, \mathbf{c}}(\cdot)$. This fixed point provides the set of all applicable rules.

It should be emphasized that using the definition of s_X , as given in Example 9, together with the rules that represent norms and policies, may cause the application of a rule in one iteration round to invalidate the application of a rule in a previous iteration round. Again, we can choose among various definitions of s_X or various classes of rules, depending on the properties that are identified as desirable by the system designer. We provide some properties that hold for the iterative version of rule application using s_X , as given in Example 9, and a certain class of norms and policies.

We assume that events (either agents' actions or actions from the environment) can change the brute as well as the institutional states of an organisation. In the following, we assume a function $\text{effect} : \xi \rightarrow 2^{I_b \cup \overline{I}_b}$ that determines the brute and communication fact consequences of an action. For example, $\text{effect}(\text{approve}(a)) = \{aa, \text{done}(\text{approve}(a))\}$ indicates that the brute effect of approving an assurance $\text{approve}(a)$ is that the assurance is approved aa and the approve action is performed, i.e., $\text{done}(\text{approve}(a))$. The processing of agents (inter)actions, which modifies the state of the organisation, is defined by the following transition rule. We also assume an update function $\otimes : 2^{I_b} \times 2^{I_b \cup \overline{I}_b} \rightarrow 2^{I_b}$ that updates a finite set of brute facts based on a finite set of brute literals. This update function can be as simple as adding positive literals and removing negative literals from the set of brute facts. For example, $\otimes(\{p\}, \{\neg p, q\}) = \{q\}$ and $\otimes(\{p\}, \{\neg p, q, r, \neg r\}) = \otimes(\{p\}, \{\neg p, q\}) = \{q\}$. Again, one may choose among various definitions of \otimes depending on some given desirable properties at hand.

Definition 11 (*Transition rules*) Let N be a finite set of norms, R a finite set of policies, \mathbf{c} a set of contexts, and **App** an application function. Also let \otimes be an update function. The following transition specifies how the organisation state changes based on event α .

$$\frac{\perp \notin \sigma'_b}{\langle \sigma_b, \sigma_i \rangle \xrightarrow{\alpha} \langle \sigma'_b, \sigma'_i \rangle} \qquad \frac{\perp \in \sigma'_b}{\langle \sigma_b, \sigma_i \rangle \xrightarrow{\alpha} \langle \sigma_b, \sigma_i \rangle} \tag{1}$$

The updated sets of brute and institutional facts (i.e., σ'_b and σ'_i , respectively) are computed in two consecutive steps through σ_b^* and σ_i^* , respectively. The computation of these sets are specified below.

1. $\sigma_b^* = \otimes(\sigma_b, effect(\alpha))$ is the update of brute facts σ_b based on the consequences of event α .
2. $\sigma_i^* = \sigma_i \setminus \{C^x(V) \mid C^y(V) \in A\} \cup A$ is the update of institutional facts with the consequences of applicable norms, where x, y are commitment states, V is the vector of arguments of the commitment, and $A = cons(App(\sigma_i \cup \sigma_b^*, N, \mathbf{c}))$.
3. $\sigma_b' = \otimes(\sigma_b^* \setminus \Pi_{act}, cons(App(\sigma_b^* \cup \sigma_i^*, R, \mathbf{c})))$ is the second (final) update of the brute facts with the consequences of applicable regulation policies. Note that communication acts Π_{act} are not stored in the set of brute fact in order to prevent repeated application of norms.
4. $\sigma_i' = \sigma_i^* \setminus cond(App(\sigma_b^* \cup \sigma_i^*, R, \mathbf{c}))$ is the set of institutional facts. Note that institutional facts for which a regulation policy is applied are removed such that a policy for an institutional fact is applied only once.

The left transition rule in (1) ensures that the performance of event α changes the state of the organisation when the triggered and applied norms are not regimented by any regulation policy, i.e., when \perp is not generated through the application of policies. By contrast, the right transition rule ensures that the state of the organisation does not change when the performance of α triggers the application of norms that are regimented by dedicated regulation policies.

According to these transition rules, the state of a multi-agent organisation can make a transition when event α is monitored, i.e., event $\alpha \in \xi$ is either an agent's action or a change in the environment. In these transition rules, the set of brute facts σ_b is first updated with the brute fact consequence of α which results in σ_b^* . Then, the set of institutional facts σ_i are updated by applying the norms to determine new commitments or change the state of the existing commitments. The state of commitments are changed by first generating new commitments from applicable norms (i.e., computing the set A of new commitments), then deleting from σ_i all commitments that are also in A (but have possibly a different state), and finally adding the newly generated commitments A to σ_i . This sequence of operations, which can change the state of the commitments, results in an update of the set of institutional facts σ_i^* . Third, the updated set of brute facts σ_b^* is updated again; this time based on the set of regulation policies and the newly generated institutional facts σ_i^* . This second update determines the brute consequences of the policies based on newly generated commitments. This results in the (final) set of brute facts σ_b' . Finally, the set of institutional facts σ_i^* is modified by removing all institutional facts for which a regulation policy has been applied. This step, which results in the (final) set of institutional facts σ_i' , prevents to apply one and the same regulation policy more than once.

The update of brute facts can be as simple as adding and removing atomic brute facts by assuming the brute consequence of events as a set of brute literals. The update of institutional facts, as defined in this transition, is also as simple as adding and removing commitments (atomic institutional facts) to/from institutional facts. In particular, new commitments, or new state of existing commitments, are computed by taking the consequence of the applicable norms. Note again that the existing commitments for which a new state is computed (i.e., $\{C^x(V) \mid C^y(V) \in A\}$) are removed from institutional facts and the result is updated with the outcome of A . This operation guarantees that the state of commitments changes according to the transitions in the commitment lifecycle depicted in Fig. 1 if we use norms as represented by the counts-as rules from Fig. 2.

We emphasize that the provided operational semantics assumes a set of norms and regulation policies, which should be designed with care and relative to this operational semantics. For example, in the final step to determine σ_i' the removal of institutional facts may have implications for the application of other regulation policies in the next transition steps. In par-

ticular, the removal of institutional facts may make policies that involve some of the removed institutional facts in their antecedents inapplicable. A reviewer of this article helpfully noted that one may want a regulation policy with the aim of changing the state of a commitment from detached to violation, by designing a policy rule with the detached commitment in the antecedent and a brute fact in the consequent. The brute fact should then lead the detached commitment to move to the violation state through a norm in the next transition step. In our view, the norm and regulation policy as sketched by the reviewer do indeed not meet the expected objective of the reviewer, although we would like to stress the suggested norm and regulation policy can be desirable if the reviewer had a different objective. For example, suppose one would like to bring a detached commitment to a violated state, unless the agents perform a repair action. The proposed norm and regulation policy ensure that the detached commitment will be moved to a violated state in the next state unless the agents perform a repair action. Again, we think that these are modelling issues and that norms and regulation policies should be designed carefully having the operational semantics in mind.

3.2.2 Properties

Given the transition rules as defined in Definition 11, the following proposition ensures that derivable transitions are sound in the sense that derivable transitions involve well-defined organisation states.

Proposition 1 *Let $\langle \sigma_b, \sigma_i \rangle$ be an organisation state, and $\langle \sigma_b, \sigma_i \rangle \xrightarrow{\alpha} \langle \sigma'_b, \sigma'_i \rangle$ a transition derived by applying the transition rules from Definition 11. We have that $\langle \sigma'_b, \sigma'_i \rangle$ is an organisation state.*

Proof Follows directly from Definition 11 and the assumptions $\otimes : 2^{\Pi_b} \times 2^{\Pi_b \cup \overline{\Pi_b}} \rightarrow 2^{\Pi_b}$ and $cons(N) \subset \Pi_i$ for a set of norms N . The assumptions ensure that $\sigma'_b \subseteq \Pi_b, \sigma'_i \subseteq \Pi_i$, and that σ'_b and σ'_i are finite sets. □

Although the above proposition establishes a general desirable property of the operational semantics, other properties may be relevant only for specific purposes or applications. For example, to formally analyse norms, policies and their consequences may require isolating the application of norms and policies in single transition steps, i.e., requiring that an agent’s action causes relevant norms and policies to be applied directly after the performance of the action. Otherwise, an agent’s action may cause norms and policies to be applied in successive transitions, possibly interfering with norms and policies that are triggered by other agents’ actions. This would make it hard, if not impossible, to properly analyse norms, policies and their consequences. The following propositions formulate some conditions under which all applicable norms and/or regulation policies are applied in one single transition step. In particular, the following proposition shows that imposing certain restriction on regulation policies ensures that all applicable regulation policies are applied in one single transition step. In the following propositions, we use $AtomA(\lambda)$ (and $AtomC(\lambda)$) to denote the set of propositional atoms involved in the antecedent (and consequent) of rules λ , which can be either norms or regulation policies.

Proposition 2 *Let $App(X, R, c)$ be defined as in Example 9 or Example 10, R be a finite set of regulation policies, and $\langle \sigma_b, \sigma_i \rangle \xrightarrow{\alpha} \langle \sigma'_b, \sigma'_i \rangle$ be a transition derived by applying the transition rules from Definition 11. Then, if $AtomA(R) \cap AtomC(R) = \emptyset$, then no regulation policy in R is applicable in organisation state $\langle \sigma'_b, \sigma'_i \rangle$, i.e.,*

$$\{(\varphi \Rightarrow_r \psi \text{ in } c) \in R \mid \sigma'_b \cup \sigma'_i \models \varphi\} = \emptyset$$

Proof Following Proposition 1 we have that $\langle \sigma'_b, \sigma'_i \rangle$ is an organisation state and following Definition 8 we have that $\sigma'_b \cup \sigma'_i$ is consistent. Moreover, the fact that $AtomA(R) \cap AtomC(R) = \emptyset$ ensures that the outcomes of the iterative and non-iterative applications of policies (Examples 9, 10) become identical. Clause 4 in Definition 11 ensures then that no regulation policy is applicable in $\langle \sigma'_b, \sigma'_i \rangle$ as it ensures that there are no institutional facts in σ'_i that can trigger a regulation policy. \square

Note that the regulation policies as presented in our running example satisfy the condition $AtomA(R) \cap AtomC(R) = \emptyset$.

Norms and regulation policies may interact in such a way that the subsequent occurrence of one and the same event (agents' actions) makes norms to become applicable in successive transitions. Therefore, the design of norms and regulation policies is crucial for the behaviour of the transition system. The following proposition shows that under specific conditions the subsequent occurrence of an event has no new effects.

Proposition 3 *Let $App(X, R, c)$ be defined as in Example 9 or Example 10, N be a set of norms, and R be a set of regulation policies. If $AtomA(\lambda) \cap AtomC(\lambda) = \emptyset$ for $\lambda = N$ or $\lambda = R$, and $AtomA(N) \cap AtomC(R) = \emptyset$ then we have:*

$$\langle \sigma_b, \sigma_i \rangle \xrightarrow{\alpha} \langle \sigma'_b, \sigma'_i \rangle \xrightarrow{\alpha} \langle \sigma''_b, \sigma''_i \rangle$$

Proof Let $\langle \sigma_b, \sigma_i \rangle \xrightarrow{\alpha} \langle \sigma'_b, \sigma'_i \rangle$ and suppose that α is applied again such that the state $\langle \sigma''_b, \sigma''_i \rangle$ is obtained, i.e., $\langle \sigma'_b, \sigma'_i \rangle \xrightarrow{\alpha} \langle \sigma''_b, \sigma''_i \rangle$. First note that the constraint $AtomA(\lambda) \cap AtomC(\lambda) = \emptyset$ for $\lambda = N$ or $\lambda = R$ ensures that the outcomes of the iterative and non-iterative applications of rules (Examples 9 and 10) become identical. Following the four clauses of the definition of transition rule in Definition 11, note that the computation of σ'_b and σ'_i results in the consequences of applicable policies from R . Then, on the second application of event α , the first two clauses of the definition the computation of σ''_b and σ''_i applies respectively the consequences of α , which have already been applied, and the applicable norms from N . However, since $AtomA(N) \cap AtomC(R) = \emptyset$, there are no applicable norms on σ'_i , and hence the second clause does not make a change. Hence $\langle \sigma''_b, \sigma''_i \rangle = \langle \sigma'_b, \sigma'_i \rangle$. \square

Note again that norms and regulation policies as presented in our running example satisfy the conditions as formulated in the above proposition.

The following example illustrates the working of the operational semantics.

Example 11 (Agents communicate) Let $\langle \{ d_0 \}, \emptyset \rangle$ be the (initial) state of the insurance organisation (c, N , and R are omitted). Suppose the Insurer communicates with the Repairer and offers him to approve the assessment of a repaired car within 4 days from the date d that the Repairer reports the repair and the insurance of the repaired car has been validated. This communication action is represented by event “offer($l, R, iv \wedge rr, aa, d, d+4$)”. In our running example, the left transition rule can be applied to derive the following transition.

$$\langle \{ d_0 \}, \emptyset \rangle \xrightarrow{offer(l,R,iv \wedge rr,aa,d,d+4)} \langle \{ d_0 \}, \{ C^c(l, R, iv \wedge rr, aa, d, d+4) \} \rangle$$

where a new commitment in conditional state is created. For this example, we assume that $effect(offer(l, R, iv \wedge rr, aa, d, d+4)) = \{ offered(l, R, iv \wedge rr, aa, d, d+4) \}$ (i.e., there is no brute fact consequence but only a communication fact consequence $offered(l, R, iv \wedge rr, aa, d, d+4)$) and d_0, d , and $d+4$ are different date propositions. The application of norms N (in the set of contexts c) to this communication fact generates a commitment (in conditional state), which is subsequently added to the set of institutional facts by means of

the update operator \otimes . Finally, the communication fact $offered(I, R, iv \wedge rr, aa, d, d+4)$ does not remain in the set of brute facts of the resulting state of the organisation.

Monitoring two subsequent but identical offers result in an organisation state that is identical to the organisation state that is achieved after monitoring the first occurrence of the offer, i.e., two subsequent but identical offers result to one and the same commitment. Two identical/equivalent offers can be made different/non-equivalent by for example using propositions with time stamps (see Sect. 4 for a more general analysis).

Example 12 (Environment changes) Let the organisation of our running example be in state $\{\{d_0\}, \{C^c(I, R, iv \wedge rr, aa, d, d+4)\}\}$ and the time elapses such that the current date d_0 of the organisation moves to the next date denoted by d_1 . This event (passing of time), represented by $d_0 \rightarrow d_1$, causes the left transition rule to apply and the following transition to be derived:

$$\begin{array}{c} \{\{d_0\}, \{C^c(I, R, iv \wedge rr, aa, d, d+4)\}\} \\ \xrightarrow{d_0 \rightarrow d_1} \\ \{\{d_1\}, \{C^c(I, R, iv \wedge rr, aa, d, d+4)\}\} \end{array}$$

We assume that propositions d_0 and d_1 cannot hold in the same state and that d_1 states occur after d_0 states. We also assume that d_1 is different to d and $d+4$ and that they do not hold in the same state. Note that the set of institutional facts is not changed as this time change does not make any norm (from Fig. 2) to become applicable.

The following example demonstrates a further execution of the multi-agent organisation as described above.

Example 13 (Agents communicate) Suppose now that on date d_1 the Repairer agent R communicates with the Insurer agent I and reports the repair of the car with a valid insurance $iv \wedge rr$. The communication actions causes the application of the left transition rule and derives the following transition.

$$\begin{array}{c} \{\{d_1\}, \{C^c(I, R, iv \wedge rr, aa, d, d+4)\}\} \\ \xrightarrow{tell(R, I, iv \wedge rr)} \\ \{\{d_1, iv \wedge rr\}, \{C^d(I, R, iv \wedge rr, aa, d, d+4)\}\} \end{array}$$

This transition takes place by applying the transition rule for events (Definition 11) due to the tell action (assuming $effect(tell(R, I, iv \wedge rr)) = \{iv \wedge rr, told(R, I, iv \wedge rr)\}$). Note that rule 2 from Fig. 2 is applied and that the communication fact $told(R, I, iv \wedge rr)$ does not remain in the set of brute facts.

The following example shows that any cancellation of a commitment before its deadline may cause the violation of the commitment and therefore the application of regulation policies.

Example 14 (Agents' communication continues) Suppose now the Insurer agent I cancels its commitment made towards the Repairer agent R to approve assessment. Assume that we have the following regulation policy indicating that a violation due to a cancellation requires the Insurer agent to pay $\$200$ to the Repairer agent, i.e.,

$$C^{vc}(I, R, iv \wedge rr, aa, d, d+4) \implies_r pay(I, R, \text{€ } 200) \text{ in } c_{10}$$

The application of the left transition rule due to this communication causes the following state transition to be derived.

$$\begin{aligned} & \langle \{d_1, iv \wedge rr\}, \{C^d(l, R, iv \wedge rr, aa, d, d+4)\} \rangle \\ & \xrightarrow{\text{cancel}(l, R, aa)} \\ & \langle \{d_1, iv \wedge rr, \text{pay}(l, R, \in 200)\}, \emptyset \rangle \end{aligned}$$

This transition takes place by again applying the transition rule for events (Definition 11) due to the cancel action (assuming that the effect of the cancel action is to cause the action proposition *cancelled* to hold, i.e., $effect(cancel(l, R, aa)) = \{cancelled(l, R, aa)\}$). Note that rule 5 from Fig. 2 is applied. The application of this norm changes the commitment state from conditional to cancellation violation, i.e., it replaces the commitment $C^c(l, R, iv \wedge rr, aa, d, d+4)$ to the commitment $C^{vc}(l, R, iv \wedge rr, aa, d, d+4)$. Moreover, the subsequent application of the policy adds the brute fact $\text{pay}(l, R, \in 200)$ to the set of brute facts and removes the commitment $C^{vc}(l, R, iv \wedge rr, aa, d, d+4)$ from the set of institutional facts.

3.3 Organisation execution

We are now ready to define the executions of an organisation. Recall that an organisation monitors the interaction among agents as well as the interaction between agents and the environment to detect norm violations and enforces norms by means of regulation policies when violations are detected. In our framework, the transition rules, presented in Definition 11, are responsible for monitor and regulation processes. The executions of an organisation, called computational runs, are determined by the transition system which is specified by the transition rules.

Definition 12 (*Computational runs*) Given the transition rules as presented in Definition 11 and an initial state of an organisation $\mathcal{O}_0 = \langle \sigma_b, \sigma_i \rangle$, a *computational run* of the organisation $CR(\mathcal{O}_0)$ is an infinite sequence $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1, \mathcal{O}_2, \alpha_2, \dots$ where $\forall_{i \geq 0} : \mathcal{O}_i \xrightarrow{\alpha_i} \mathcal{O}_{i+1}$ is a transition derivable by applying a transition rule and α_i is an event occurring in state \mathcal{O}_i .

Possible computational runs of an organisation are due to possible sequences of occurred events (i.e., agents' actions or changes in the environment). In other words, the interactions between agents and between agents and the environment determine the computational runs of an organisation.

Proposition 4 Let $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1, \mathcal{O}_2, \alpha_2, \dots$ be a computational run of an organisation. We have that \mathcal{O}_n is an organisation state for $n \geq 0$, i.e., $\forall n \geq 0 : \mathcal{O}_n = \langle \sigma_b^n, \sigma_i^n \rangle$ with $\sigma_b^n \subseteq \Pi_b, \sigma_i^n \subseteq \Pi_i$.

Proof By induction. The initial state $\mathcal{O}_0 = \langle \sigma_b^0, \sigma_i^0 \rangle$ is an organisation state, i.e., $\sigma_b^0 \subseteq \Pi_b$ and $\sigma_i^0 \subseteq \Pi_i$. Suppose $\mathcal{O}_k = \langle \sigma_b^k, \sigma_i^k \rangle$ is an organisation state for $k \geq 0$. Proposition 1 ensures that $\mathcal{O}_k = \langle \sigma_b^k, \sigma_i^k \rangle \xrightarrow{\alpha_k} \langle \sigma_b^{k+1}, \sigma_i^{k+1} \rangle = \mathcal{O}_{k+1}$ is an organisation state, i.e., $\sigma_b^{k+1} \subseteq \Pi_b, \sigma_i^{k+1} \subseteq \Pi_i$, and σ_b^{k+1} and σ_i^{k+1} are finite. □

We give some example execution traces corresponding to agent actions in the insurance scenario. The context used in the traces is the combined context of Figs. 2, 3, and 4. Specific predicates correspond to agent actions in the scenario, such as *insurance-validated*.

Trace 1 Agent actions are manifest in communication actions, and in the impact of the actions upon the commitments in the institutional state. The trace in Table 1 begins with the *offers* that create the five commitments. Column Rule gives the Counts-as Rule in the notation $X.Y$ for rule Y of Figure X. We abbreviate the logical propositions by their initials, e.g., *pp* for premium-paid, and we do not show deadlines.

Once the commitments are established, in step 8, the customer proves payment of the insurance. In steps 10 and 11, she reports car damage. In step 15, the garage has completed the repair. In step 19 the garage receives confirmation of the approval and in step 20 it tells the customer the car is ready, within the 10 days promised to the customer. Payment to the garage by the insurer concludes the scenario.

Trace 2 Table 2 takes the same scenario but supposes that the assessor is delayed in his work. According to commitment C_3 , the Assessor commits to having the assessment done no later than two days after the Repairer informs the Assessor of a report of a damaged car and a validated insurance on that car. The Assessor does not complete the assessment for 4 days after it began. This means the deadline on C_3 is violated (step 13). The Assessor is motivated to inform the Insurer of completion of the assessment, because otherwise he will not be paid by the Insurer for his work. We suppose (as in Table 1) that the Assessor informs the Insurer the same day that he completes the assessment. It takes the Insurer up to 4 days to approve the assessment (C_{1b}) and thus the Repairer does not receive confirmation that the assessment is approved until day 21. This means that, although the Repairer has completed the repair work on the car, it cannot release the car to the Customer until day 22. This violates its deadline on C_4 , which was due by day 20.

We suppose a penalty clause in the insurance contract, whereby the Insurer guarantees that properly-reported eligible car damage will be repaired within 10 days. One could model the guarantee by an additional commitment from Insurer to Customer (not shown); the deadline on the consequent in C_4 derives from this. As we have it the guarantee is modelled in C_4 and the following regulative rule:

$$C_4^{vi} \implies_r \text{pay}(\text{Insurer}, \text{Customer}, \text{€ } 200)$$

which is applied for any delay beyond the deadline. This rule models the penalty clause upon the Insurer which states that the Insurer will compensate the Customer for delayed repairs by the amount of 200 euro. Thus, according to this rule in step 21, payment is made by the Insurer to the Customer.

4 Logical analysis of interaction protocols

Many different kinds of norms and regulation policies can not only be defined in our formal framework, but also be compared and analysed. As we allow the user of our formal framework to define her own set of norms and regulations, and small changes in these norms and regulations may have a big effect on the system's behaviour, it is important to also provide formal methods to analyse the behaviour of the system. In this section we show how to analyse a normative system by proving various properties, and we illustrate these properties on the three contexts defined in the previous sections. Note the difference with the properties we defined in the previous section: thus far, all properties held for all normative systems, as they were properties of our operational semantics. The properties discussed in this section hold only for a particular normative system.

Table 1 Execution trace 1 in car insurance scenario

Step	Agent	Communication act	Day	Rule	Institutional state	Brute facts
0	-	-	0	-	{}	{}
1	I	<i>offer</i> (I, R, iv \wedge rr, aa)	1	2.1	{ C^c_{1a} }	{}
2	I	<i>offer</i> (I, R, iv \wedge rr, pd)	2	2.1	{ C^c_{1a}, C^c_{1b} }	{}
3	I	<i>offer</i> (I, A, ad, ap)	3	2.1	{ $C^c_{1a}, C^c_{1b}, C^c_2$ }	{}
4	A	<i>offer</i> (A, R, dr \wedge iv, ad)	4	2.1	{ $C^c_{1a}, C^c_{1b}, C^c_2, C^c_3$ }	{}
5	R	<i>offer</i> (R, C, iv \wedge cd, cr)	5	2.1	{ $C^c_{1a}, C^c_{1b}, C^c_2, C^c_3, C^c_4$ }	{}
6	I	<i>offer</i> (I, C, pp, iv)	6	2.1	{ $C^c_{1a}, C^c_{1b}, C^c_2, C^c_3, C^c_4, C^c_5$ }	{}
7	C	<i>pp</i>	7	3.1	{ $C^c_{1a}, C^c_{1b}, C^c_2, C^c_3, C^c_4, C^d_5$ }	{ <i>pp</i> }
8	I	<i>iv</i>	8	3.2	{ $C^c_{1a}, C^c_{1b}, C^c_2, C^c_3, C^c_4, C^c_5$ }	{ <i>iv, pp</i> }
9	-	-	9	-	{ $C^c_{1a}, C^c_{1b}, C^c_2, C^c_3, C^c_4, C^c_5$ }	{ <i>cd, iv, pp</i> }
10	C	<i>tell</i> (C, A, iv \wedge cd)	10	2.2	{ $C^c_{1a}, C^c_{1b}, C^c_2, C^c_3, C^d_4, C^c_5$ }	{ <i>cd, iv, pp</i> }
11	C	<i>tell</i> (C, R, iv \wedge cd)	10	2.2	{ $C^c_{1a}, C^c_{1b}, C^c_2, C^c_3, C^d_4, C^c_5$ }	{ <i>cd, iv, pp</i> }
12	R	<i>tell</i> (R, A, dr \wedge iv)	11	2.2	{ $C^c_{1a}, C^c_{1b}, C^c_2, C^c_3, C^d_4, C^c_5$ }	{ <i>cd, dr, iv, pp</i> }
13	A	<i>ad</i>	13	3.2	{ $C^c_{1a}, C^c_{1b}, C^c_2, C^c_3, C^d_4, C^c_5$ }	{ <i>ad, cd, dr, iv, pp</i> }
14	A	<i>tell</i> (A, I, ad)	13	2.2	{ $C^c_{1a}, C^c_{1b}, C^d_2, C^c_3, C^d_4, C^c_5$ }	{ <i>ad, cd, dr, iv, pp</i> }
15	R	<i>rr</i>	15	3.1	{ $C^c_{1a}, C^c_{1b}, C^c_2, C^c_3, C^d_4, C^c_5$ }	{ <i>ad, cd, dr, iv, pp, rr</i> }
16	R	<i>tell</i> (R, I, iv \wedge rr)	15	2.2	{ $C^d_{1a}, C^d_{1b}, C^c_2, C^c_3, C^d_4, C^c_5$ }	{ <i>ad, cd, dr, iv, pp, rr</i> }
17	I	<i>ap</i>	17	2.4	{ $C^d_{1a}, C^d_{1b}, C^c_2, C^c_3, C^d_4, C^c_5$ }	{ <i>ad, ap, cd, dr, iv, pp, rr</i> }
18	I	<i>tell</i> (I, A, ap)	17	2.4	{ $C^d_{1a}, C^d_{1b}, C^c_2, C^c_3, C^d_4, C^c_5$ }	{ <i>ad, ap, cd, dr, iv, pp, rr</i> }
19	I	<i>tell</i> (I, R, aa)	19	2.4	{ $C^s_{1a}, C^d_{1b}, C^c_2, C^c_3, C^d_4, C^c_5$ }	{ <i>aa, ad, ap, cd, dr, iv, pp, rr</i> }
20	R	<i>tell</i> (R, C, cr)	20	2.2	{ $C^s_{1a}, C^d_{1b}, C^c_2, C^c_3, C^d_4, C^c_5$ }	{ <i>aa, ad, ap, cr, dr, iv, pp, rr</i> }
21	I	<i>pd</i>	22	3.2	{ $C^s_{1a}, C^s_{1b}, C^c_2, C^c_3, C^d_4, C^c_5$ }	{ <i>aa, ad, ap, cr, dr, iv, pd, pp, rr</i> }

Table 2 Execution trace 2 in car insurance scenario

Step	Agent	Communication act	Day	Rule	Institutional state	Brute facts
0	-	-	0	-	{}	{}
1	I	<i>offer</i> (I, R, $iv \wedge ir$, aa)	1	2.1	{ C_{1a}^i }	{}
2	I	<i>offer</i> (I, R, $iv \wedge ir$, pd)	2	2.1	{ C_{1a}^i, C_{1b}^i }	{}
3	I	<i>offer</i> (I, A, ad, ap)	3	2.1	{ $C_{1a}^i, C_{1b}^i, C_2^i$ }	{}
4	A	<i>offer</i> (A, R, $dr \wedge iv$, ad)	4	2.1	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i$ }	{}
5	R	<i>offer</i> (R, C, $iv \wedge cd$, cr)	5	2.1	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{}
6	I	<i>offer</i> (I, C, pp , iv)	6	2.1	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>pp</i> }
7	C	<i>pp</i>	7	3.1	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>iv, pp</i> }
8	I	<i>iv</i>	8	3.2	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>cd, iv, pp</i> }
9	-	-	9	-	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>cd, iv, pp</i> }
10	C	<i>tell</i> (C, A, $iv \wedge cd$)	10	2.2	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>cd, iv, pp</i> }
11	C	<i>tell</i> (C, R, $iv \wedge cd$)	10	2.2	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>cd, iv, pp</i> }
12	R	<i>tell</i> (R, A, $dr \wedge iv$)	11	2.2	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>cd, dr, iv, pp</i> }
13	-	-	14	4.1	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>ad, cd, dr, iv, pp</i> }
14	A	<i>ad</i>	15	3.2	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>ad, cd, dr, iv, pp</i> }
15	A	<i>tell</i> (A, I, ad)	15	2.2	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>ad, cd, dr, iv, pp</i> }
16	R	<i>rr</i>	15	3.1	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>ad, cd, dr, iv, pp, rr</i> }
17	R	<i>tell</i> (R, I, $iv \wedge rr$)	15	2.2	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>ad, cd, dr, iv, pp, rr</i> }
18	I	<i>ap</i>	19	2.4	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>ad, ap, cd, dr, iv, pp, rr</i> }
19	I	<i>tell</i> (I, A, ap)	19	2.4	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>ad, ap, cd, dr, iv, pp, rr</i> }
20	-	-	21	4.1	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>ad, ap, cd, dr, iv, pp, rr</i> }
21	I	<i>pay</i> (Insurer, Customer, € 200)	21	N/A	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>ad, ap, cd, dr, iv, pp, rr</i> }
22	I	<i>tell</i> (I, R, aa)	21	2.4	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>aa, ad, ap, cd, dr, iv, pp, rr</i> }
23	R	<i>tell</i> (R, C, cr)	22	2.2	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>aa, ad, ap, cr, dr, iv, pp, rr</i> }
24	I	<i>pd</i>	24	3.2	{ $C_{1a}^i, C_{1b}^i, C_2^i, C_3^i, C_4^i, C_5^i$ }	{ <i>aa, ad, ap, cr, dr, iv, pd, pp, rr</i> }

Bold values indicate a delay by the agents completing their actions compared to the days they should have completed their actions

More precisely, the properties in this section assume, besides Definitions 1–12, also Action Set 1, contexts $\mathbf{c} = \{c_1, c_2, c_3\}$, the set of commitment states S occurring in these contexts of \mathbf{c} , App and s_X as defined in Example 9, and moreover the following two reasonable constraints on the update function and the regulation policy:

- The update function \otimes in Definition 11 is based on the fact that only one communication proposition is true at a time, and
- Only vc and vt commitments occur in the antecedents of a regulation policy in Definition 6.

In the first item, the fact that only one communication proposition is true at a time together with the fact that only bidirectional communication is considered so far may suggest that our framework is restricted to one bidirectional communication at a time, and that our framework cannot model simultaneous communication between multiple agents with possibly interfering effects. However, this is not the case as our framework can model such cases by assuming concurrent (communication) actions in the transition rules as proposed in Definition 11, i.e., by assuming α to be a concurrent (communication) action. Of course, it should also be assumed that the effect function provides the brute effect of simultaneous communication actions with their interfering effects. In the rest of the article, we will not discuss this issue in detail as this is not the focus of our work.

4.1 Temporal properties

We start with *temporal properties* describing the dynamics of the commitment states. These properties illustrate that the commitment states follow the commitment lifecycle in Fig. 1. Together they imply continuity properties about reaching termination states. Consequently, a conditional and detached commitment state will lead to precisely one termination state. First, each commitment is in at most one state.

Proposition 5 (Uniqueness) *For all computational runs $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots$ and all $x \neq y$, and all $\phi_x, \phi_y, \delta_y, \delta_x$, we have at each state $\mathcal{O}_k = \langle \sigma_b, \sigma_i \rangle$ that σ_i contains at most one of: $\mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, $\mathbf{C}^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, $\mathbf{C}^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, $\mathbf{C}^{vc}(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, $\mathbf{C}^{vt}(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, $\mathbf{C}^e(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, and [4] $\mathbf{C}^t(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$.*

Proof By induction over the sequence of states in a computational run. The induction predicate $P(k)$ is that at state \mathcal{O}_k , at most one of the given commitments is in the institutional facts σ_i . Base condition. Initially, the set of commitments is empty. Thus $P(0)$ holds. Induction step. Assume $P(k)$, the induction predicate holds at state \mathcal{O}_k . Given the constraint on the update function \otimes , at most one of the speech act or action propositions Π_a is in the updated brute facts $\sigma_b^* = \otimes(\sigma_b, \text{effect}(\alpha))$. Consequently, given that x is unequal to y , at most one counts-as rule in context $\{c_1, c_2, c_4\}$ is applicable (as can directly be verified by the rules in Figs. 2, 3, 4). Given that at new state σ_i' of the computational run, the old commitments are removed, at moment $k + 1$, at most one of the commitments is in the institutional facts σ_i . Thus $P(k + 1)$ holds. By induction principle, the induction predicate holds at each moment of the computational run. \square

Although it may seem obvious that a commitment can be in one state only, it may easily occur that a less careful representation of the counts-as rules leads to a state in which two commitments are derived from one. Even with the given rules in context $\{c_1, c_2, c_4\}$, this may happen when we replace App as defined in Example 9 by the one in Example 10,

where we may apply rules iteratively. To see this, let $\mathcal{O}_k = \langle \sigma_b^k, \sigma_i^k \rangle$ and consider the case of an offer, of which the deadline is already true: $\alpha_k = offer(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, and $\sigma_b^k \cup \sigma_i^k \models \neg\phi_y \wedge \delta_y$. Then we derive two commitments $\mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ and $\mathbf{C}^l(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$. In other words, if we want to use the **App** from Example 10, and we want to have the uniqueness property of commitments, we need to change the count-as rules.

In the rest of this section, we write $\mathcal{O}_k \models \phi$ as an abbreviation of $\sigma_b^k \cup \sigma_i^k \models \phi$.

Besides the assumptions made above, the remaining properties in this section are based in addition on the following assumption of non-repetition.

Definition 13 (*Non-repetition*) A computational run $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots$ satisfies the *non-repetition property* if and only if there are no two actions α_k and α_l for $k \neq l$ in the computational run that are the same offer $\alpha_k = \alpha_l = offer(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$.

In particular, if all offers of the computational run have a unique identifier as part of ϕ_x and ϕ_y , then the non-repetition property is satisfied. Thus, if we assume that each $offer(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ is done with unique ϕ_y and ϕ_x , then there is no repetition.

Second, after the temporal properties, we state continuity properties about some of the termination states: once they are true, they will stay true. This does not hold for the violation states, as the violation will be removed once the sanction is applied (i.e., once the regulation policy is implemented).

Proposition 6 For all computational runs $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots$, which satisfy the non-repetition property, if $\mathbf{C}^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, $\mathbf{C}^e(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, or $\mathbf{C}^l(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ is in σ_i of state $\mathcal{O}_k = \langle \sigma_b, \sigma_i \rangle$, then it is also in σ'_i of state $\mathcal{O}_l = \langle \sigma'_b, \sigma'_i \rangle$ for $k < l$. Moreover, if the set of regulation policies is empty, then the same property holds also for $\mathbf{C}^{vc}(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ and $\mathbf{C}^{vt}(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$.

Proof Suppose that one of $\mathbf{C}^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, $\mathbf{C}^e(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, or $\mathbf{C}^l(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ is in σ_i of state $\mathcal{O}_k = \langle \sigma_b, \sigma_i \rangle$. Then we have that $\alpha_l = offer(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ for some $l < k$. Consequently, due to the non-repetition property in Definition 13, we have $\forall l > k, \alpha_l \neq offer(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$. In addition, due to the fact that a commitment can only be in one state at the time, i.e., Proposition 5, we have that none of the counts-as rules in context $\{c_1, c_2, c_4\}$ is applicable (as can directly be verified by the rules in Figs. 2, 3, 4). Thus $\mathbf{C}^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, $\mathbf{C}^e(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, or $\mathbf{C}^l(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ is also in σ'_i of state $\mathcal{O}_l = \langle \sigma'_b, \sigma'_i \rangle$ for $k < l$. The same argument goes through for $\mathbf{C}^{vc}(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ and $\mathbf{C}^{vt}(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ provided there are no regulation policies. □

Third, we give continuity properties about conditional and detached commitments: they hold until they are fulfilled or the deadline is reached. We use the weak until **U** operator of linear time logic (LTL) to express the properties compactly: ϕ holds until ψ if and only if ϕ is true until the first position in which ψ is true (or forever if such a position does not exist).

Proposition 7 We say that a property $\phi\mathbf{U}\psi$ holds at state \mathcal{O}_k in computational run $CR(\mathcal{O}_0) = \mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots$, written as $CR(\mathcal{O}_0), \mathcal{O}_k \models \phi\mathbf{U}\psi$, if and only if for all positions $j \geq k$ we have

- if for all i such that $k \leq i \leq j$ we have $\mathcal{O}_i \not\models \psi$, then $\mathcal{O}_j \models \phi$.

If the computational run is clear from context, as below, we also write $\mathcal{O}_k \models \phi\mathbf{U}\psi$ for $CR(\mathcal{O}_0), \mathcal{O}_k \models \phi\mathbf{U}\psi$.

The following properties hold at state $\mathcal{O}_k = \langle \sigma_b^k, \sigma_i^k \rangle$ of a computational run satisfying the non-repetition property:

- If $\alpha_k = offer(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, then
 - $\mathcal{O}_{k+1} \models \mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \mathbf{U}(\mathbf{C}^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \vee \mathbf{C}^e(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \vee \mathbf{C}^f(x, y, \phi_y, \phi_x, \delta_y, \delta_x)).$
- If $\alpha_k = tell(y, x, \phi_y)$ or α_k is an action, and $\mathcal{O}_k \models \mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ and $\sigma_b^{k*} = \otimes(\sigma_b^k, effect(\alpha_k)) \models \neg\delta_y \wedge \phi_y$ and is consistent, then
 - $\mathcal{O}_{k+1} \models \mathbf{C}^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \mathbf{U}(\mathbf{C}^{vc}(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \vee \mathbf{C}^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \vee \mathbf{C}^f(x, y, \phi_y, \phi_x, \delta_y, \delta_x)).$

Proof - Assume $\alpha_k = offer(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$. Due to the first counts-as rule, and the assumption that $\sigma_b^{k*} = \otimes(\sigma_b^k, effect(\alpha_k))$ is consistent for communication acts, we have $\mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in σ_i^{k+1} for σ_i^{k+1} of state $\mathcal{O}_{k+1} = \langle \sigma_b^{k+1}, \sigma_i^{k+1} \rangle$. In addition, due to the fact that a commitment can only be in one state at the time, i.e., Proposition 5, we have that at most one of the counts-as rules in context $\{c_1, c_2, c_4\}$ is applicable (as can directly be verified by the rules in Figs. 2, 3, 4), and these rules lead to $\mathbf{C}^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ or $\mathbf{C}^e(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ or $\mathbf{C}^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ or $\mathbf{C}^f(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$.

- Assume $\alpha_k = tell(j, i, \phi_y)$ or α_k is an action, assume $\mathcal{O}_k \models \mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ and assume that $\sigma_b^{k*} = \otimes(\sigma_b^k, effect(\alpha_k)) \models \neg\delta_y \wedge \phi_y$ and is consistent. Due to the second counts-as rule in context c_1 , or the first rule in context c_2 , and the second and third condition of the property, we have $\mathbf{C}^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in σ_i^{k+1} for σ_i^{k+1} of state $\mathcal{O}_{k+1} = \langle \sigma_b^{k+1}, \sigma_i^{k+1} \rangle$. Again, due to the fact that a commitment can only be in one state at the time, i.e., Proposition 5, we have that at most one of the counts-as rules in context $\{c_1, c_2, c_4\}$ is applicable (as can directly be verified by the rules in Figs. 2, 3, 4). These rules lead to one of $\mathbf{C}^{vc}(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ or $\mathbf{C}^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ or $\mathbf{C}^f(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$. □

If we change the temporal properties or the rules, then also the properties of the operational semantics will change. For example, if we adopt context where a conditional commitment cannot be directly satisfied without going through a detached state, we have the following variant:

- If $\alpha_k = offer(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, then
 - $\mathcal{O}_{k+1} \models \mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \mathbf{U}(\mathbf{C}^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \vee \mathbf{C}^e(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \vee \mathbf{C}^f(x, y, \phi_y, \phi_x, \delta_y, \delta_x)).$

Further temporal properties can be defined to illustrate that the operational semantics behaves according to the commitment lifecycle in Fig. 1. For example, we may wonder whether we have that a deadline succeeds in removing commitments. As above, this depends on the precise specification of the counts-as rules. Here we have to be careful about the borderline conditions. For example, consider an offer, of which the deadline is already true. For such an offer, we may check whether the commitment is created at the next moment:

- If $\alpha_k = offer(x, y, \phi_y, \phi_x, \top, \top)$, then $\mathcal{O}_{k+1} \models \mathbf{C}^c(x, y, \phi_y, \phi_x, \top, \top)$

or whether the commitment is terminated:

- If $\alpha_k = offer(x, y, \phi_y, \phi_x, \top, \top)$, then $\mathcal{O}_{k+1} \models \mathbf{C}^f(x, y, \phi_y, \phi_x, \top, \top)$

Likewise, we might check that no commitment holds when the deadline is true:

- $\mathcal{O}_k \models \delta_y \rightarrow \neg\mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$
- $\mathcal{O}_k \models \delta_x \rightarrow \neg\mathbf{C}^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$

4.2 Interference

In this subsection we consider whether one commitment lifecycle can affect another one; we call this *interference*. The first proposition shows that one action can lead to moves in multiple commitment lifecycles.

Proposition 8 *If the organisation updates the institutional facts based on a performed communication action $\text{tell}(y, x, p)$ or an action $\text{done}(p)$, then it can detach or satisfy multiple commitments at once.*

Proof Consider the following example. Assume the organisation updates the institutional facts based on the following four actions, where $s_{(x,y)}$ stands for one service of y to x , $t_{(x,y)}$ for another one, and $p_{(y,20)}$ stands for paying twenty euro. $\text{offer}(x, y, s_{(x,y)}, p_{(y,20)}, t_2, t_5)$, $\text{offer}(x, y, t_{(x,y)}, p_{(y,20)}, t_2, t_5)$, $\text{done}(s_{(b,i)})$, and $\text{done}(t_{(b,i)})$. This leads to two detached commitments, $\mathbf{C}^d(x, y, s_{(x,y)}, p_{(y,20)}, t_2, t_5)$ and $\mathbf{C}^d(x, y, t_{(x,y)}, p_{(y,20)}, t_2, t_5)$. Performing action $\text{done}(p_{(y,20)})$ will now move both detached commitments to satisfied commitments, i.e., agent x de-commits itself by paying $\text{€}20$ instead of $\text{€}40$. \square

Example 15 Another similar example is when an agent makes two offers that differ only in the deadlines. In that case, the commitment with the earlier deadline ‘subsumes’ the one with the later deadline.

These examples illustrates that the agents must be careful to distinguish propositions. In this case, the agents should syntactically distinguish $p_{(y,20)}$ referring to pay $\text{€}20$ in $\text{offer}(x, y, s_{(x,y)}, p_{(y,20)}, t_2, t_5)$ and $\text{offer}(x, y, t_{(x,y)}, p_{(y,20)}, t_2, t_5)$ respectively, and it is left to the agent to prevent this undesired interference.

The second proposition shows that there may be two commitments which can be satisfied individually, but not together. At first sight it may seem that this can be done simply by commitments for p and for $\neg p$, but these two commitments can be satisfied consecutively.

Proposition 9 *There may be commitments which can be satisfied individually, but not together.*

Proof Consider the following example. If you offer to do p before deadline q and you offer to do q before deadline p , we have a contradiction (with the interpretation of deadline as in context c_3). \square

We may define interference in terms of independence of commitment lifecycles: two commitment lifecycles are independent if any kind of interleaving of the two lifecycles leads to the same final state. We leave the formalisation of these notions to further research.

4.3 Redundancy

Broersen and van der Torre [8] define norms to be *redundant* if their removal from the normative system does not change the set of detached institutional facts. In the current article, the commitments correspond to these institutional facts. They distinguish various kinds of redundancy properties depending on various kinds of equivalence among computational runs. In this subsection, we consider redundancy properties that do not change the brute facts. We confine ourselves to the following regulation policy, where agents are blacklisted once they violated a commitment, i.e., which treats all violations in the same way.

Definition 14 (*Blacklisting regulation policy*) A *blacklisting regulation policy* is a regulation policy consisting of all instances of the following two policies:

$$\begin{aligned} C^{vt}(x, y, \phi_y, \phi_x, \delta_y, \delta_x) &\Rightarrow_r add_{x,BL} \\ C^{vc}(x, y, \phi_y, \phi_x, \delta_y, \delta_x) &\Rightarrow_r add_{x,BL} \end{aligned}$$

In the following we define a partial run as an initial segment of a computational run. We say that an offer is *blacklist redundant* in a partial run if the computational run without the offer and the computational run with the offer lead to the same brute facts in the computational run. This implies that the two computational runs have violations in the same states.

Definition 15 A partial run is a finite sequence $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots, \mathcal{O}_n$ where \mathcal{O}_i is a multi-agent organisation state, α_i is the action performed in state \mathcal{O}_i , and $\forall i \geq 0 : \mathcal{O}_i \xrightarrow{\alpha_i} \mathcal{O}_{i+1}$ is a transition derived by applying the given transition rules.

An offer α is *blacklist redundant* in a partial run if and only if for all computational runs extending the partial run, $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots, \mathcal{O}_n, \alpha_n, \mathcal{O}_{n+1}, \dots$, we have that the computational run $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots, \mathcal{O}_n, \alpha, c, \alpha_n, \mathcal{O}'_{n+1}, \dots$ is such that $\sigma_k^b = \sigma_k^{b'}$ for all $k > n$, where $\mathcal{O}_k = (\sigma_k^b, \sigma_k^i)$ and $\mathcal{O}'_k = (\sigma_k^{b'}, \sigma_k^{i'})$.

The following proposition gives, given a partial run, some offers that are blacklist redundant. The properties are very natural and may be taken as requirements on a normative system. They all say, in some way, that it is redundant to offer less than what you already offered in the past. The fourth property has a more technical nature as it considers actions to make a deadline true, but such mixtures of actions and deadlines give important insights in the working of the normative system.

Proposition 10 *If the following conditions hold:*

- R contains blacklisting regulation policy, and does not contain any other rules than the blacklisting regulation policy,
- $\otimes(\sigma_b, effect(\alpha)) \setminus \Pi_{act} = \sigma_b$ for communication actions α , i.e., communication actions do not have side effects, and
- The update function \otimes is successful in the sense that the update with the regulation policy $\otimes(\sigma_b^* \setminus \Pi_{act}, cons(\mathbf{App}(\sigma_b^* \cup \sigma_i^*, R, \mathbf{c})))$ contains conclusions $cons(\mathbf{App}(\sigma_b^* \cup \sigma_i^*, R, \mathbf{c}))$, and
- The update function \otimes is blacklist persistent in the sense that once an agent is blacklisted, it will remain blacklisted,

Then the following properties hold:

- If $\mathcal{O}_n \models C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, then $offer(x, y, \phi_y \wedge \psi, \phi_x, \delta_y, \delta_x)$ is blacklist redundant in $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots, \mathcal{O}_n$. In other words, if x has offered y that if y does ϕ_y before δ_y , then x will do ϕ_x before δ_x , then it is redundant for x to offer y that if y does $\phi_y \wedge \psi$ then x will do ϕ_x , under the same deadlines.
- If $\mathcal{O}_n \models C^c(x, y, \phi_y, \phi_x \wedge \psi, \delta_y, \delta_x)$, then $offer(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ is blacklist redundant in $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots, \mathcal{O}_n$. In other words, if x has offered y that if y does ϕ_y before δ_y , then x will do $\phi_x \wedge \psi$ before δ_x , then it is redundant for x to offer y that if y does ϕ_y , then x will do ϕ_x , under the same deadlines.
- If $\mathcal{O}_n \models C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ and $\mathcal{O}_n \models C^c(x, y, \psi_y, \phi_x, \delta_y, \delta_x)$, then $offer(x, y, \phi_y \vee \psi_y, \phi_x, \delta_y, \delta_x)$ is blacklist redundant in $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots, \mathcal{O}_n$. In other words, if x has offered y that if y does ϕ_y before δ_y , then x will do ϕ_x before δ_x , and

x has offered *y* that if *y* does ψ_y , then *x* will do ϕ_x under the same deadlines, then it is redundant for *x* to offer *y* that if *y* does $\phi_y \vee \psi_y$ then *x* will do ϕ_x , under the same deadlines.

- If $\mathcal{O}_n \models \mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ and $\mathcal{O}_n \models \mathbf{C}^c(x, y, \phi_y, \delta_x, \delta_y, \delta'_x)$, then $\text{offer}(x, y, \phi_y, \phi_x, \delta_y, \delta'_x)$ is blacklist redundant in $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots, \mathcal{O}_n$. In other words, if *x* has offered *y* that if *y* does ϕ_y before δ_y , then *x* will do ϕ_x before δ_x , and *x* has offered *y* that if *y* does ϕ_y before the same deadline, then *x* will do δ_x before δ'_x , then it is redundant for *x* to offer *y* that if *y* does ϕ_y then *x* will do ϕ_x before δ'_x .

Proof – Assume a partial run $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots, \mathcal{O}_n$ s.t. $\mathcal{O}_n \models \mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, and $\alpha = \text{offer}(x, y, \phi_y \wedge \psi, \phi_x, \delta_y, \delta_x)$. If $\mathcal{O}_n = \langle \sigma_b, \sigma_i \rangle$, then $\mathcal{O}_{n+1} = \langle \sigma_b, \sigma_i \cup \mathbf{C}^c(x, y, \phi_y \wedge \psi, \phi_x, \delta_y, \delta_x) \rangle$ due to the constraint on the update function. Now due to the occurrence of ϕ_x in both commitments, and the fact that we can satisfy a commitment even if it is not detached yet, if the first is satisfied the second is too. Moreover, if the second one is violated, the first one is violated too. The result follows.

- Assume $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots, \mathcal{O}_n$ such that $\mathcal{O}_n \models \mathbf{C}^c(x, y, \phi_y, \phi_x \wedge \psi, \delta_y, \delta_x)$, and $\alpha = \text{offer}(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$. If $\mathcal{O}_n = \langle \sigma_b, \sigma_i \rangle$, then, due to the constraint on the update function, $\mathcal{O}_{n+1} = \langle \sigma_b, \sigma_i \cup \mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \rangle$. Now if the first is satisfied the second is too. Moreover, due to the occurrence of ϕ_y in both commitments, if the second one is violated, the first one is violated too. The result follows.
- Assume a partial run $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots, \mathcal{O}_n$ s.t. $\mathcal{O}_n \models \mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, $\mathcal{O}_n \models \mathbf{C}^c(x, y, \psi_y, \phi_x, \delta_y, \delta_x)$ and $\alpha = \text{offer}(x, y, \phi_y \vee \psi_y, \phi_x, \delta_y, \delta_x)$. If $\mathcal{O}_n = \langle \sigma_b, \sigma_i \rangle$, then $\mathcal{O}_{n+1} = \langle \sigma_b, \sigma_i \cup \mathbf{C}^c(x, y, \phi_y \vee \psi_y, \phi_x, \delta_y, \delta_x) \rangle$ due to the constraint on the update function. Now due to the occurrence of ϕ_x in both commitments, and the fact that we can satisfy a commitment even if it is not detached yet, if the first ones are satisfied the third one is too. Moreover, if the third one is detached, one of the first two is detached too. And finally, if the third one is violated, then one of the first two is violated too. The result follows.
- Assume a partial run $\mathcal{O}_0, \alpha_0, \mathcal{O}_1, \alpha_1 \dots, \mathcal{O}_n$ s.t. $\mathcal{O}_n \models \mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$, $\mathcal{O}_n \models \mathbf{C}^c(x, y, \phi_y, \delta_x, \delta_y, \delta'_x)$ and $\alpha = \text{offer}(x, y, \phi_y, \phi_x, \delta_y, \delta'_x)$. If $\mathcal{O}_n = \langle \sigma_b, \sigma_i \rangle$, then $\mathcal{O}_{n+1} = \langle \sigma_b, \sigma_i \cup \mathbf{C}^c(x, y, \phi_y, \phi_x, \delta_y, \delta'_x) \rangle$ due to the constraint on the update function. Now due to the temporal sequence in the commitments, and the fact that we can satisfy a commitment even if it is not detached yet, if the first ones are satisfied the third one is too. Moreover, if the third one is detached, the first two are detached too. And finally, if the third one is violated, then one of the first two is violated too. The result follows. □

The following proposition shows an offer which may seem blacklist redundant at first sight, but that is not blacklist redundant under our present assumptions.

Proposition 11 *If $\mathbf{C}^c(x, y, p, q, d_1, d_2) \wedge \mathbf{C}^c(x, y, p, r, d_1, d_2)$, then $\text{offer}(x, y, p, q \wedge r, d_1, d_2)$ is not necessarily blacklist redundant. If *x* has offered *y* that if *y* does *p* then *x* will do *q*, and if *y* does *p*, then *x* will do *r*, then it is not redundant to offer *y* that *x* will do $q \wedge r$, under the same conditions.*

Proof (Sketch) Since *q* and *r* may be true at different timepoints, the commitment for $q \wedge r$ cannot be derived from it. □

5 Norms for alternative commitment lifecycles

In this section we discuss some possible sets of norms that specify different commitment lifecycles. Our objective in particular is to show how our methodology is not tied to a particular view of commitments, but can accommodate alternative views. We leave detailed application of our methodology to alternative norm sets for the future.

5.1 Truthfulness of communication

The norms in Fig. 2 require truthfulness of the communication actions. This requirement is encoded by including the proposition that is involved in the speech act in the antecedent of the counts-as rules. For example, the second norm, rule 2, changes the state of a commitment from conditional to detached when an agent y has told agent x that ϕ_y is true and moreover ϕ_y is true (we ignore the deadline proposition). The inclusion of both communication actions and factual statements check whether the communication actions are performed truthfully. If we do not include this check, then we have instead the norms in Fig. 5.

The norms of Fig. 5 are appropriate in situations where agents trust other agents sufficiently to believe what others tell them, and in situations where agents are not able to check the veracity of what they are told. By contrast, the norms of Fig. 2 are more appropriate in situations where agents do not ‘blindly’ trust other agents.

5.2 Deadlines

As explained in Sect. 3.1.1, a commitment may change its state from detached to violated if the environment make a transition (time elapses) through which the deadline (a brute fact) becomes true. Figure 6 lists alternative examples of norms that govern the dynamics of commitments due to the internal mechanism of the environment. The difference with context c_3 is the borderline case where the actions is satisfied precisely at the deadline.

- The elapse of deadline δ_x counts as the violation of a detached commitment. The application of this rule 1 in Fig. 6 removes the detached commitment from institutional facts and adds a corresponding violated commitment to it.

- 1 $offered(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \Rightarrow_n C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_4
- 2 $told(y, x, \phi_y) \wedge C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg\delta_y \Rightarrow_n C^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_4
- 3 $told(x, y, \phi_x) \wedge C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg\delta_y \Rightarrow_n C^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_4
- 4 $told(x, y, \phi_x) \wedge C^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg\delta_x \Rightarrow_n C^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_4
- 5 $cancelled(x, y, \phi_x) \wedge C^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg\delta_x \Rightarrow_n C^{vc}(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_4
- 6 $failed(y, x, \phi_y) \wedge C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg\delta_y \Rightarrow_n C^e(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_4
- 7 $cancelled(x, y, \phi_x) \wedge C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg\delta_y \Rightarrow_n C^t(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_4
- 8 $released(y, x, \phi_y) \wedge C^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg\delta_x \Rightarrow_n C^t(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_4

Fig. 5 Example context 4 (c_4). Norms based on communication actions without factual check

- 1 $C^d(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \delta_x \wedge \neg\phi_x \Rightarrow_n C^{vt}(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_5
- 2 $C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \delta_y \wedge \neg\phi_y \Rightarrow_n C^e(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_5

Fig. 6 Example context 5 (c_5). Counts-as rules specify the lifecycle of commitments based on environment events. Commitments can be fulfilled before or at the deadline

- 1 $offered(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \Rightarrow_n C^P(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_6
- 2 $accepted(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge C^P(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \neg\phi_x \Rightarrow_n C^c(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ in c_6
- 3 $C^x(x, y, \phi_y, \phi_x, \delta_y, \delta_x) \wedge \phi_x \Rightarrow_n C^s(x, y, \phi_y, \phi_x, \delta_y, \delta_x)$ for $x \in \{p, c, d\}$ in c_6

Fig. 7 Example context 6 (c_6). Counts-as rules specify the lifecycle of commitments based on communication actions

- The elapse of deadline δ_y counts as expiration of a conditional commitment. The application of this rule 2 in Fig. 6 removes the conditional commitment from institutional facts and adds a corresponding expired commitment to it.

5.3 Explicit acceptance of commitments

Recall that the commitment lifecycle given in Sect. 2.2 follows that of Singh [62,65], in which commitments are unilateral, not requiring acceptance by the other party (but therefore not binding upon them).

The rules in Sect. 3.1.1, given in Figs. 2 and 3, thus model the specific commitment lifecycle as presented in Fig. 1. Different rules can be proposed for alternative lifecycles. For example, following Castelfranchi [21], one can require that conditional commitments are created only after explicit acceptance by the creditor. Or, if the content of what is committed by a debtor is satisfied before the commitment is accepted by the creditor or before it is detached for the debtor, or after its detachment, then the commitment will be considered as satisfied.

These alternative rules are presented in Fig. 7. In particular, rule 1 indicates that the proposed commitments are represented by C^P). Rule 2 indicates that a proposed commitment becomes a conditional commitment after explicit acceptance by the creditor. Finally, rule 3 in Fig. 7 indicates that a proposed, conditional, and detached commitment is considered as being satisfied when the debtor’s commitment is satisfied.

5.4 Further topics

There are many other ways in which the commitment lifecycles can be changed or extended. We mention two of such changes here, but we do not present the counts-as rules.

First, in our operational semantics for commitments, we declare that a commitment is satisfied if its consequent is true, irrespective of whether the antecedent has been made true yet or not. This is expressed in rule 1 in Fig. 2. A different semantics for commitments would say that a commitment is satisfied only if its antecedent is made true and then its consequent is made true. A modification of rules 3 and 4 in Fig. 2 can express this. Specifically, we can simply remove rule 3.

Second, commitments as modelled in this article consider achievement, whereas we can also define rules for persistence. For example, if x commits that p will *not* hold until deadline d , then we have the following. First, if the deadline is achieved, then the prohibition is satisfied. Second, if d is false at some moment before the deadline, then the commitment is violated. We can define this in our framework by defining other constitutive rules.

For example, let d_0 be the start of the year and d_1 be the end of the year. The Insurer will increase the no-claims bonus of the Owner, if the latter does not submit an insurance claim that year. We have a rule such as:

$$\begin{aligned} & \{ \{ d_0 \}, \{ C^c(I, O, \neg\text{insurance-claim, increase-no-claims-bonus, } d_0, d_1) \} \} \\ & \xrightarrow{d_0 \rightarrow d_1} \\ & \{ \{ d_1 \}, \{ C^d(I, O, \neg\text{insurance-claim, increase-no-claims-bonus, } d_0, d_1) \} \} \end{aligned}$$

6 Related work and discussion

Billhardt et al. [7] summarize *agreement technologies* as comprising of two main elements: “(1) a normative context, that determines the rules of the game ... and (2) a call-by-agreement interaction method, where an agreement for action between the agents that respects the normative context is established first; then actual enactment of the action is requested.” Our work speaks to both elements, through our emphasis of the normative organisational context, on the one hand, founded on social commitments, on the other.

Social commitments are now well established in the literature, and various lifecycles for commitments have been proposed (e.g., [42, 51, 52]). The relation between communication and commitments has been examined from several perspectives. Among the first to study commitments and Agent Communication Languages (ACLs) were Singh [62] and Colombetti [27]. Singh, for example, develops a formal logical semantics for an ACL based on communication actions interpreted in terms of commitments, using Computation Tree Logic. de Boer et al. [35] were also among the first to define an operational semantics for an ACL, and in the same year, McBurney et al. did so for an agent interaction protocol [53].

The idea of non-communicative actions and the enforcement of sanctions by monitoring the state of commitments became common in the agents literature. The schema of our model is standard when dealing with commitments in agents organisations: organisations consist of facts, norms, commitments and sanction, the agents can perform a set of actions which lead to the creation of commitments, and these commitments have rules in form of count-as rules and when the commitments are violated, sanctions apply [4, 9, 20, 24, 25, 42, 43, 49, 51, 52, 55].

Moreover, as with our work, formalisations are often based on a lifecycle for commitments using an operational semantics. However, the aim is typically to give a semantics for a large variety of communication actions, e.g. for propose or request, and such works therefore have to define their semantics in considerable detail. Further still, the trend is to make the languages ever more expressive but hence complex, for example by introducing higher order commitments [61], meta-commitments [68], embedded temporal regulations [4], organisational goals, other norms besides those commitment-based norms, and so on.

Building on earlier work [27, 42], Fornara et al. [43] propose an ACL based on communication actions seen as *instrumental actions*, i.e., actions that modify reality such as changing the physical environment, when bound to *institutional actions*, i.e., actions that change institutional attributes. They define norms as “rules that manipulate commitments of the agents engaged in an interaction” and thus, unlike our work, define norms as event-driven rules and provide a more limited operational semantics. In this and earlier work (e.g., [42]), Fornara and colleagues adopt a state-based definition of commitments that includes an *unset* state (to express directive communicative acts like requests). They allow temporal propositions in the content of a commitment, but have a timeout only for commitments in *unset* state.

Flores et al. [41] propose an ACL founded on four levels: compositional, conversational, commitment state, and joint activity. Like Fornara et al., these authors define message semantics (at one level) based on commitments. They require explicit acceptance of a commitment by the creditor (or debtor). Our focus by contrast is on communication and sanction within an operationalizable organisational setting.

In contrast to these and similar works, Kibble [49] critiques ACLs founded on speech acts, preferring an extension of commitment-based protocols [4] incorporating Brandom’s notions of commitment, entitlement, and perspectival commitment stores. He considers both ‘active’ acceptance (e.g., of a request) and ‘passive’ acceptance (by silence). Similarly, Boella et al. [9] contrast ACLs based on (private) mental attitudes (e.g., BDI models) such as the FIPA

standard [40], and ACLs based on social commitments, arguing that the latter solve many but not all of the problems identified with the former. The authors suggest a role-based ontology of *public* mental attitudes that distinguishes *action commitments* and *propositions*.

It is not for us to enter in this article the debate in the commitments literature whether or not explicit acceptance should be part of the construct.³ Following Singh and colleagues (e.g., Chopra and Singh [25]), the commitment lifecycle in Sect. 2.2 does not require explicit acceptance. In contrast to the above works, our stated purpose is not to define an ACL but to establish how norms are to be operationalized in an organisational setting. As we demonstrated in Sect. 5, our methodology can be applied for alternative sets of communication actions than those selected in Sect. 3.1.1 and following, and for alternative sets of commitment-based norms.

Singh [61] is an example of commitment management within an organisational structure, by means of social policies defined as higher-order commitments. Going further, Venkatraman and Singh [68] define *metacommitments* as first-order commitments, and use them to define protocols in which messages between agents correspond to operations on commitments. They verify adherence to these commitment-based protocol using temporal logic. While metacommitments can be seen as a type of norm, the authors do not consider sanctions for violated protocols.

Baldoni et al. [4] extend earlier work on commitment-based protocols by distinguishing constitutive rules which define the meaning of actions and regulative rules which define the flow of execution. Despite giving a language for representing regulative rules based on constraints among commitments, the authors do not consider norms and sanctions.

Formal verification of commitments is exemplified by El Menshawey et al. [37], who define a temporal logic and use symbolic model checking. Such formal model-checking approaches have been applied to (pre-defined) commitment-based protocols. Günay et al. [46] propose an algorithm for agents to generate novel protocols at runtime.

Chesani et al. [22,23] were perhaps the first to address commitment monitoring through a formal yet operational approach. They define a declarative, practical language for commitments, and exhibit a proof approach with the reactive event calculus. Notably, their language includes deadlines, and compensation actions for events such as for deadline violation. They treat metric time and commitments with data parameters, which we do not. Our work takes the approach of a rule-based semantics rather than a logic programming-based event calculus approach, and provides a methodology that applies to different commitment lifecycles.

Knobbout and Dastani [50] consider norms as directed to coalitions of agents and provide a logic to reason about the impact of such norms on the overall multi-agent system behaviours under various norm compliance assumptions. In this paper, a norm states that a set of agents should not take certain actions in specific states. The authors introduce different types of norm compliance behaviours such as a coalition of agents obey/violate norms that are directed to precisely this coalition or a it and all its subcoalitions. The proposed logic can be used to reason if a coalition of agents can achieve their own objectives by behaving according to a specific norm compliance type when the system is populated by other agents that behave according to another specific norm compliance type.

Bulling and Dastani [19] approach norms from a mechanism design perspective and focus on whether the enforcement of a specific set of norms by means of sanctions can influence the behaviour of agents and align them with the objectives of the system designer. The authors assume that the agents are rational in the sense that they behave according to their own

³ We are grateful for a reviewer pointing out that most legal jurisdictions construe commitments as arising only when accepted by someone, not when first uttered: see [57].

preferences. In this paper, norms and preferences are represented as LTL formulae, and the agents behaviours are studied from a game theoretic perspective. The proposed framework can be used to determine whether a set of norms with their corresponding sanctions can implement the objectives of the system designer, defined as social choice function, in Nash equilibria. The framework can also be used to synthesize norms and sanctions such that their enforcement implement the objectives of the system designer in Nash equilibria.

Compared to earlier work, our approach differs in the following regards. First, we use counts-as rules explicitly as technical constructs while Fornara et al. [42], for instance, treat counts-as relation primarily as linguistic conventions. Note that counts-as rules in our work can be seen as defeasible rules. Second, we provide an operational semantics for interactions (among which communication actions) within an organisational setting. Fornara et al. provide semi-formal specification of organisations and consider only communication actions. Further, we analyze the properties of interaction within an organisational setting. Third, we consider the effect of non-communicative actions as well as the elapse of deadlines. Fourth we have sanction rules while earlier works leave open the question of what should happens when commitments are violated. Fifth, in contrast to some works, we adopt a contemporary lifecycle of commitments, following Chopra and Singh [25] and precedents (see earlier discussion in this section).

These last authors are interested in simplifying the specification of commitment-based protocols or requirements. Our aim is in line with the latter paper [25], and the two approaches are orthogonal. Chopra and Singh capture business requirements with commitment-based specifications, and then group these specifications into reusable methods. They consider protocol enactment but not the organisational setting, which is our focus. By contrast, we do not aim to directly model business requirements, but start from the lifecycle of commitments and define a minimal language to make all possible changes to the commitment base.

An alternative approach that aims for the same level of genericity as we pursue, is the approach of using deontic-inspired specifications which are monitored and enforced. An example of this kind of approach is the work of Felipe et al. [39]. A discussion for the future is to compare our commitment-based approach versus a deontic-inspired approach.

In earlier work, Singh examined when a commitment may be considered satisfied but some change or event can lead to checking whether it is still valid, and transitioning the commitment to a new state based on whether it is valid or not [71]. Instead, we model violation in terms of regulative rules, which are separate from the rules modelling the commitment lifecycle.

Pasquier and Chaib-draa [55] study the link between communication based on social commitments, and the internal intentions of the agents involved. Similarly, Winikoff [70] develops a mapping from commitments to BDI-style plans. Concerned with operationalizing inter-agent interaction, these latter works do not consider institutional aspects.

Alechina et al. [1] are among those who ask how norm-aware agents can be implemented. They provide a norm-aware agent programming language that allows agents to deliberate on their goals, norms and sanctions before deciding which plan to select and execute.

Carabelea and Boissier [20] do consider an organisational setting and model expected behaviour of agents playing specified roles using commitments. They posit an “Institutional Agent” that rewards or sanctions agents that fulfil or deviate from specified behaviour. They do not consider communication actions, nor present an operational semantics.

Marengo et al. [52] develop a formal semantics of commitments with embedded temporal regulations, and study important notions of control and safety. They argue that their framework accommodates both constitutive and regulative norms “grounded in communication”. In their approach, which is not directly founded on communication actions, regulations are reified into the commitment. By contrast, in an organisational setting it is the organisation

that judges compliance to commitments, according to norms that are specified separately, and, further, may impose sanctions.

Singh studies norms for governance in the context of socio-technical systems [63]. His notion of ‘governance’ is related to regulation in an organizational setting, but differs from traditional organizational environments such as we have studied them, in that governance is “the administration of such a system by the stakeholders themselves.” Singh characterizes a lifecycle for norms where the terminal state is computed based on a truth table of the antecedent and consequent.

Chopra et al. [24] are interested in modelling exchange of services in a distributed setting. They model service-oriented application protocols in terms of messages and their effect on commitments. Whereas, Chopra and Singh [25] are interested in simplifying the specification of commitment-based protocols or requirements. To be explored is how our approach can be combined with these works to aid specifying agent and organisation behaviour in realistic settings.

7 Future work

This article considered two principal communication actions: *offer* and *tell*. Alternative semantics can be defined for these actions using other constitutive rules, and compared to the ones we defined. Our methodological approach will still be applicable. Moreover, our model can be extended to a wider range of communication actions and provide their semantics in terms of social commitments. Extending the set of communication actions should be done carefully since commitments may create unwanted interferences.

Second, at a technical level, certain assumptions and modelling choices can be examined further. For example, the content of communication actions and commitments are assumed to be identical. This limits the framework to model only cases where single communication actions, and never several together, should lead to state transition, e.g., *offer*($p \wedge q$) and *tell*(p), *tell*(q).

Third, commitments, as we have adopted them, possess special slots for deadlines. Alternatively, as proposed by Marengo et al. [52], the deadline could be part of the propositional content (the antecedent and consequent) of the commitment. In that case, the deadline would not have a special status. For example, we could express the commitment that the insurer will have paid the assessment within 7 days, and (further) approved it within 14 days. Whether such composite commitments can be expressed by several commitments in our language, and in general the advantages and disadvantages of the two definitions from the point of view of the organisational setting, is left for further research.

Fourth, we considered here generic counts-as rules that were designed based on the semantics of specific communication actions (i.e., *offer* and *tell*). A possible extension would be to investigate the interaction between such generic counts-as rules and domain-specific counts-as rules representing domain-specific norms. For example, we can consider norms such as ‘buying a book while having no money on a credit card counts-as a violation’. Among the questions that arise are such as when an agent without money on its credit card then offers another agent to buy its book, i.e., creating a commitment that it cannot fulfil without violating the norm. It is not clear how such a commitment and violation should interact and what would be outcome of such interactions, e.g., should the organisation decide to change the state of the commitment to a violated state?

Fifth, the lifecycle of commitments does not include the notion of delegation of a commitment from one agent to another [48]. For instance, in the scenario, if the car cannot be repaired by the repair garage, perhaps a more specialized garage for the model of car would have the expertise. The transference of the car to a specialized garage occurs with the delegation of commitment C_4 from Repairer to Repairer2.

Finally, the dynamics of organisations is an important topic which we have not treated here. We believe that representing and analysing mechanisms in our framework to dynamically change the rules of the norms and regulation policies at run-time would be an important issue to explore.

8 Summary

Multi-agent systems in an organisational setting typically possess an entity defining interaction protocols, monitoring agent interaction, and enforcing interaction norms by means of regulation policies. We call this entity the ‘organisation’ or ‘institution’. For our purposes, an organisation consists of two main processes: the *monitoring process* checks for conformance of agents’ behaviour to protocols or norms, while the *enforcement process* ensures the coordination by means of implementing the regulation policies. We are indifferent to whether the organisation is either a virtual or a real organisation [56].

In this article, we propose to model the interaction protocols of the organisation by explicitly-represented interaction norms. This norm representation has two advantages. The first advantage is that we can operationalise the enforcement of protocols by means of norm enforcement, and the second advantage is that we can analyse the protocol by a logical analysis of the norms. To make this idea more precise, we adopt insights and methods from commitment-based approaches as well as from norm-based approaches.

First, we show how to use explicitly represented norms to model variety in the commitment lifecycle. As far as the organisation component is concerned, agent interactions affect the state of the institutional facts, in particular the commitments, whereas regulation policy is responsible for updating the organisation state as a consequence of detected violations. By modelling the interaction protocols of the organisation as a set of explicitly represented norms, to which the agents’ interaction should adhere, we are able to generalise interaction protocols to represent, for example, which agents can detach or satisfy commitments, whether obligations can be satisfied before they are detached or not, and temporal conditions such as whether a commitment which is fulfilled exactly at the moment of the deadline, counts as violated or not.

Second, we introduce an operational semantics to operationalise protocol enforcement. In this way, we show how to operationalise the enforcement of protocols by means of norm enforcement. An organisational state consists of a set of brute facts, a set of institutional facts, a set of contexts, a set of norms and a set of regulation policies. Given a way to define the set of applicable rules in a context, two transition rules define how to update the brute and institutional facts. In the same way, we could add rules for changing the contexts and the sets of norms and policies.

Third, we show how to analyse interaction protocol of the organisation and thus commitment cycles by logically analysing the norms. This also gives us a way to logically analyse the enforcement of the protocol. Since the norms are represented explicitly, and the change depends on the norms only, we can use techniques and insights from deontic logic [44]. Temporal properties show that we can formally check that the operational semantics corresponds

to the commitment lifecycle. Moreover, we can analyse interference between distinct commitments, which may lead to problems in protocol analysis. Finally, redundancy of norms tells us whether the protocol behaves as expected. For example, if one makes an offer, it is redundant to make the same offer again, or to make an offer which is strictly worse for the other party.

Hence, the contribution of the article is the systematic methodology to formally model and analyse commitment lifecycles in terms of norms and regulation policies that govern the creation and dynamics of commitments emerged during agents' interaction. Our proposed framework is based on a generic operational semantics that operationalize the enforcement of norms and regulation policies, and thereby operationalize the dynamics of commitments. Our framework is generic in the sense that it allows formal instantiation and comparison of different approaches to commitment dynamics.

The model, semantics and analysis have been illustrated by a running example from a vehicle insurance domain. The sample execution traces showed the monitoring of agent communication and non-communication actions, and the application of regulation policies.

Throughout the article, we have emphasised the methodological innovation of our approach. We address the usual questions regarding agent communication, coordination and monitoring, and we adopt the usual formal methods like constitutive norms. However, the way we establish our formal framework is different from all other work, as we do not constrain ourselves to one particular commitment lifecycle. Of course, this does not mean that any set of norms and policies defined in our framework will behave well. It is part of our methodology that the behaviour of the organisation is left to the person, who defines the norms and policies. The properties show that the commitment cycle we use in the main part of our discussion, consisting of contexts $\{c_1, c_2, c_3\}$, is well behaved. Whether other norms or policies are as well behaved has to be checked independently.

Acknowledgements Thanks to the JAAMAS reviewers who helped us improve the article. We thank the participants of the COIN'12 workshop at AAMAS'12 for the discussions of the preliminary idea of the article [34], and for suggestions that have directed our investigations. Further preliminary presentation of this work was made at the AAMAS'12 conference [33]. Thanks to Amit Chopra, Steven McNamara and Paolo Torroni. NYS Acknowledges Award Number 102853 from the University Research Board, American University of Beirut, and thanks the Operations group at the Cambridge Judge Business School and the fellowship at St Edmund's College, Cambridge.

References

1. Alechina, N., Dastani, M., & Logan, B. (2012). Programming norm-aware agents. In *Proceedings of AAMAS'12* (pp. 1057–1064).
2. Alechina, N., Dastani, M., & Logan, B. (2013). Reasoning about normative update. In *IJCAI 2013, Proceedings of the 23rd international joint conference on artificial intelligence*. Beijing, August 3–9, 2013.
3. Andrighetto, G., Governatori, G., Noriega, P., & van der Torre, L. (Eds.). (2013). *Normative multi-agent systems, dagstuhl follow-ups* (Vol. 4). Dagstuhl: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
4. Baldoni, M., Baroglio, C., & Marengo, E. (2010). Behavior-oriented commitment-based protocols. In *Proceedings of ECAI'10* (pp. 137–142).
5. Baldoni, M., Boella, G., & van der Torre, L. (2006). Roles as a coordination construct: Introducing powerJava. *Electronic Notes in Theoretical Computer Science*, 150(1), 9–29.
6. Baldoni, M., Boella, G., & van der Torre, L. (2009). The interplay between relationships, roles and objects. In *Proceedings of 3rd international conference on fundamentals of software engineering (FSEN'09)* (pp. 402–415)

7. Billhardt, H., Centeno, R., Cuesta, C. E., Fernández, A., Hermoso, R., Ortiz, R., et al. (2011). Organisational structures in next-generation distributed systems: Towards a technology of agreement. *Multiagent and Grid Systems*, 7(2–3), 109–125.
8. Boella, G., Broersen, J., & van der Torre, L. (2008). Reasoning about constitutive norms, counts-as conditionals, institutions, deadlines and violations. In *Proceedings of PRIMA'08* (pp. 86–97).
9. Boella, G., Damiano, R., Hulstijn, J., & van der Torre, L. (2007). A common ontology of agent communication languages: Modelling mental attitudes and social commitments using roles. *Applied Ontology*, 2(3–4), 217–265.
10. Boella, G., Hulstijn, J., & van der Torre, L. (2005). Interaction in normative multi-agent systems. *Electronic Notes in Theoretical Computer Science*, 141(5), 135–162.
11. Boella, G., & van der Torre, L. (2004). Regulative and constitutive norms in normative multiagent systems. In *Proceedings of KR'04* (pp. 255–266).
12. Boella, G., & van der Torre, L. (2006). Coordination and organization: Definitions, examples and future research directions. *Electronic Notes in Theoretical Computer Science*, 150(3), 3–20.
13. Boella, G., & van der Torre, L. (2007). The ontological properties of social roles in multi-agent systems: Definitional dependence, powers and roles playing roles. *Artificial Intelligence and Law*, 15(3), 201–221.
14. Boella, G., & van der Torre, L. (2008). Substantive and procedural norms in normative multiagent systems. *Journal of Applied Logic*, 6(2), 152–171.
15. Boella, G., van der Torre, L., & Verhagen, H. (2006). Introduction to normative multiagent systems. *Computational & Mathematical Organization Theory*, 12(2–3), 71–79.
16. Boella, G., van der Torre, L., & Verhagen, H. (2008). Introduction to the special issue on normative multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 17(1), 1–10.
17. Broersen, J. M., Dastani, M., Hulstijn, J., Huang, Z., & van der Torre, L. (2001). The BOID architecture: Conflicts between beliefs, obligations, intentions and desires. In *Proceedings of 5th international conference on autonomous agents (Agents'01)* (pp. 9–16).
18. Bulling, N., & Dastani, M. (2011). Verifying normative behaviour via normative mechanism design. In *Proceedings of IJCAI'11* (pp. 103–108).
19. Bulling, N., Dastani, M., & Knobbout, M. (2013). Monitoring norm violations in multi-agent systems. In *Proceedings of AAMAS'13* (pp. 491–498).
20. Carabelea, C., & Boissier, O. (2006). Coordinating agents in organizations using social commitments. *Electronic Notes in Theoretical Computer Science*, 150(3), 73–91.
21. Castelfranchi, C. (1995). Commitments: From individual intentions to groups and organizations. In *Proceedings of 1st international conference on multiagent systems (ICMAS'95)* (pp. 41–48).
22. Chesani, F., Mello, P., Montali, M., & Torroni, P. (2009). Commitment tracking via the reactive event calculus. In *Proceedings of IJCAI'09* (pp. 91–96).
23. Chesani, F., Mello, P., Montali, M., & Torroni, P. (2013). Representing and monitoring social commitments using the event calculus. *Autonomous Agents and Multi-Agent Systems*, 27(1), 85–130.
24. Chopra, A. K., Dalpiaz, F., Giorgini, P., & Mylopoulos, J. (2010). Modeling and reasoning about service-oriented applications via goals and commitments. In *Proceedings of CAiSE'10* (pp. 113–128).
25. Chopra, A. K., & Singh, M. P. (2011). Specifying and applying commitment-based business patterns. In *Proceedings of AAMAS'11* (pp. 475–482).
26. Coffee, J. C., Jr, & Sale, H. A. (2012). *Securities regulation* (12th ed.). Eagan, MN: Foundation Press.
27. Colombetti, M. (2000). A commitment-based approach to agent speech acts and conversations. In *Proceedings of workshop on agent languages and communication policies* (pp. 21–29).
28. Dastani, M. (2009). Normative multi-agent organizations. In *Engineering societies in the agents world X, LNCS* (Vol. 5881, pp. 247–249). Heidelberg: Springer.
29. Dastani, M., Arbab, F., & de Boer, F. S. (2005). Coordination and composition in multi-agent systems. In *Proceedings of AAMAS'05* (pp. 439–446).
30. Dastani, M., Meyer, J. C., & Grossi, D. (2013). A logic for normative multi-agent programs. *Journal of Logic and Computation*, 23(2), 335–354.
31. Dastani, M., Meyer, J. C., & Tinnemeier, N. A. M. (2012). Programming norm change. *Journal of Applied Non-Classical Logics*, 22(1–2), 151–180.
32. Dastani, M., Tinnemeier, N., & Meyer, J. J. C. (2009). A programming language for normative multi-agent systems. In *Multi-agent systems: Semantics and dynamics of organizational models*. Hershey, PA: Information Science Reference.
33. Dastani, M. van der Torre, L., & Yorke-Smith, N. (2012). A programming approach to monitoring commitments in an organisational environment. In *Proceedings of AAMAS'12* (pp. 1373–1374).
34. Dastani, M., van der Torre, L., & Yorke-Smith, N. (2013). Monitoring interaction in organisations. In *Coordination, organizations, institutions, and norms in agent systems VIII. LNCS* (Vol. 7756, pp. 17–34). Heidelberg: Springer.

35. de Boer, F. S., van Eijk, R., van der Hoek, W., & Meyer, J. J. (2003). A fully abstract model for the exchange of information in multi-agent systems. *Theoretical Computer Science*, 290(3), 1753–1773.
36. Dechesne, F., di Tosto, G., Dignum, V., & Dignum, F. (2013). No smoking here: Values, norms and culture in multi-agent systems. *Artificial Intelligence and Law*, 21(1), 79–107.
37. El Menshawhy, M., Bentahar, J., Qu, H., & Dssouli, R. (2011). On the verification of social commitments and time. In *Proceedings of AAMAS'11* (pp. 483–490).
38. Esparcia, S., Argente, E., Centeno, R., & Hermoso, R. (2011). Enhancing MAS environments with organizational mechanisms. *International Journal on Artificial Intelligence Tools*, 20(4), 663–690.
39. Felipe, L. O., Álvarez-Napagao, S., & Vázquez-Salceda, J. (2012). Towards a framework for the analysis of regulative norm performance in complex networks. In *Proceedings of 1st international conference on agreement technologies (AT'12)* (pp. 103–104).
40. FIPA. (2002). Communicative act library specification. Retrieved October 21, 2015, from www.fipa.org/specs/fipa00037/SC00037J.html.
41. Flores, R. A., Pasquier, P., & Chaib-draa, B. (2007). Conversational semantics sustained by commitments. *Autonomous Agents and Multi-agent Systems*, 14, 165–186.
42. Fornara, N., & Colombetti, M. (2004). A commitment-based approach to agent communication. *Applied Artificial Intelligence*, 18(9–10), 853–866.
43. Fornara, N., Viganò, F., & Colombetti, M. (2007). Agent communication and artificial institutions. *Autonomous Agents and Multi-agent Systems*, 14, 121–142.
44. Gabbay, D., Horty, J., Parent, X., van der Meyden, R., & van der Torre, L. (Eds.). (2013). *Handbook of deontic logic and normative systems*. London: College Publications.
45. Grossi, D., Dignum, F., Dastani, M., & Royakkers, L. M. M. (2005). Foundations of organizational structures in multiagent systems. In *Proceedings of AAMAS'05* (pp. 690–697).
46. Günay, A., Winikoff, M., & Yolum, P. (2015). Dynamically generated commitment protocols in open systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 29(2), 192–229.
47. Jonge, D. D., Rosell, B., & Sierra, C. (2013). Human interactions in electronic institutions. In *Proceedings of 2nd international conference on agreement technologies (AT'13)* (pp. 75–89).
48. Kafali, O., & Torroni, P. (2011). Social commitment delegation and monitoring. In J. Leite, P. Torroni, T. Gotnes, G. Boella, & L. van der Torre (Eds.), *Computational logic in multi-agent systems, LNCS* (Vol. 6814, pp. 171–189). Berlin: Springer.
49. Kibble, R. (2006). Speech acts, commitment and multi-agent communication. *Computational and Mathematical Organization Theory*, 12, 127–145.
50. Knobbout, M., & Dastani, M. (2012). Reasoning under compliance assumptions in normative multiagent systems. In *Proceedings of AAMAS'12* (pp. 331–340).
51. Mallya, A. U., Yolum, P., & Singh, M. P. (2003). Resolving commitments among autonomous agents. In *Proceedings of workshop on agent communication, LNCS* (Vol. 2922, pp. 166–182). Heidelberg: Springer.
52. Marengo, E., Baldoni, M., Baroglio, C., Chopra, A. K., Patti, V., & Singh, M. P. (2011). Commitments with regulations: Reasoning about safety and control in REGULA. In *Proceedings of AAMAS'11* (pp. 467–474).
53. McBurney, P., van Eijk, R., Parsons, S., & Amgoud, L. (2003). A dialogue game protocol for agent purchase negotiations. *Journal of Autonomous Agents and Multi-Agent Systems*, 7(3), 235–273.
54. Ohlbach, H. J., & Gabbay, D. (1998). Calendar logic. *Journal of Applied Non-Classical Logics*, 8(4), 291–323.
55. Pasquier, P., & Chaib-draa, B. (2006). Integrating social commitment-based communication in cognitive agent modeling. In *Agent communication II, LNCS* (Vol. 3859, pp. 76–92). Heidelberg: Springer.
56. Raffaini, M. (2001). The virtual organisation. In *Proceedings of 7th international conference on concurrent enterprising (ICE'01)*.
57. Reinach, A. (1913). *Jahrbuch für Philosophie und phänomenologische Forschung*, chap. Die apriorischen Grundlagen des bürgerlichen Rechtes (Vol. 1, pp. 685–847). Halle: M. Niemeyer.
58. Riemsdijk, M. B. V., Hindriks, K. V., & Jonker, C. M. (2009). Programming organization-aware agents. In H. Aldewereld, V. Dignum, & G. Picard (Eds.), *Engineering societies in the agents world X, LNCS* (Vol. 5881, pp. 98–112). Heidelberg: Springer.
59. Searle, J. (1995). *The construction of social reality*. New York, NY: The Free Press.
60. Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language*. Cambridge: Cambridge University Press.
61. Singh, M. P. (1997). Commitments among autonomous agents in information-rich environments. In *Proceedings 8th european workshop on modelling autonomous agents in a multi-agent world* (pp. 141–155).
62. Singh, M. P. (2000). A social semantics for agent communication languages. In *Issues in agent communication 2000, LNCS 1916* (pp. 31–45). Heidelberg: Springer.

63. Singh, M. P. (2013). Norms as a basis for governing sociotechnical systems. *ACM Transactions on Intelligent Systems and Technology*, 5(1), 21.
64. Taylor, F. W. (1911). *The principles of scientific management*. New York, NY: Harper & Brothers.
65. Telang, P. R., & Singh, M. P. (2012). Specifying and verifying cross-organizational business models. *IEEE Transactions on Services Computing*, 5, 305–318.
66. Tinnemeier, N. A. M., Dastani, M., & Meyer, J. C. (2009). Roles and norms for programming agent organizations. In *Proceedings of AAMAS'09* (pp. 121–128).
67. van der Torre, L. W. N., Hulstijn, J., Dastani, M., & Broersen, J. M. (2004). Specifying multiagent organizations. In *Proceedings of 7th international workshop on deontic logic in computer science (DEON'04)* (pp. 243–257).
68. Venkatraman, M., & Singh, M. P. (1999). Verifying compliance with commitment protocols. *Autonomous Agents and Multi-Agent Systems*, 2(3), 217–236.
69. Weber, M. (Ed.). (1947). *The theory of social and economic organization*. New York, NY: The Free Press. Translated by A.M. Henderson and T. Parsons.
70. Winikoff, M. (2007). Implementing commitment-based interactions. In *Proceedings of AAMAS'07* (pp. 873–880).
71. Xing, J., & Singh, M. P. (2003). Engineering commitment-based multiagent systems: A temporal logic approach. In *Proceedings of AAMAS'03* (pp. 891–898).