Baldoni, Matteo; Baroglio, Cristina; Marengo, Elisa; Patti, Viviana.
Constitutive and Regulative Specifications of Commitment Protocols: a Decoupled Approach (Extended Abstract), in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, AAAI Press / International Joint Conferences on Artificial Intelligence, 2015, 978-1-57735-738-4, pp: 4143-4147.

Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

Constitutive and Regulative Specifications of Commitment Protocols: A Decoupled Approach (Extended Abstract)*

Matteo Baldoni(1), **Cristina Baroglio**(1), **Elisa Marengo**(2), **Viviana Patti**(1)
(1) Università degli Studi di Torino, Dip. di Informatica, Torino, Italy
{matteo.baldoni,cristina.baroglio,viviana.patti}@unito.it
(2) Faculty of Computer Science, Free University of Bozen-Bolzano, Bolzano, Italy
elisa.marengo@unibz.it

Abstract

We analyze the emerging trends from research on multi-agent interaction protocols, on workflows and on business processes. We propose a definition of commitment-based interaction protocols, characterized by the decoupling of the *constitutive* and the *regulative* specifications, where the latter is explicitly represented based on *constraints among commitments*. The 2CL language, for writing regulative specifications, is also presented.

1 Introduction

The term "interaction protocol" refers to a pattern of behavior that allows a set of agents to become a multi-agent system (MAS), by engaging expected cooperations. Particularly relevant are *commitment protocols* [Singh, 2000; Yolum and Singh, 2001; 2002]. Commitments are literals that can hold in the social state of the system, representing the fact that a debtor commits to a creditor to bring about some condition. All agents using a commitment protocol share the semantics of a set of actions which affect the social state. The greatest advantages of commitment protocols, with respect to other approaches to interaction, are that they *do not overconstrain* the specification by imposing unnecessary orderings on the execution of the shared actions, and that by giving a public and agreed meaning to the social actions, they allow working on actual knowledge rather than on beliefs about each others' mental states, preserving the agents' autonomy. Nonetheless, commitment protocols *do not yet suit well* those situations where the evolution of the social state is constrained by conventions, laws, preferences or habits, because they do not allow the specification of legal patterns of interaction. We tackle this problem by adopting Searle's [1995] distinction between the *constitutive* and *regulative* specifications of interaction. Constitutive rules, by identifying certain behaviors as foundational of a certain type of activity, create that activity. Regulative rules contingently constrain a previously constituted activity. They rule the "flow of activity", by capturing some important characteristics of how things should be carried on in *specific contexts* of interaction [Cherry, 1973].

*This paper is an extended abstract of the ACM Trans. on Int. Sys. and Tech. publication [Baldoni *et al.*, 2013].

A clear separation of the constitutive from the regulative specification would bring many advantages, mostly as direct effects of the obtained *modularity*: *easier re-use* of actions in different contexts, *easier customization* on the protocol, *easier composition* of protocols. As a consequence, MAS would gain greater *openness*, *interoperability*, and *modularity of design*. Interoperability would be better supported because it would be possible to verify it with respect to specific aspects (e.g. at the level of actions [Chopra and Singh, 2008; Chopra, 2009; Chopra and Singh, 2009] or at the level of regulative rules [Baldoni *et al.*, 2011b]). Protocols would be more open in the sense that their modularity would allow designers to easily adapt them to different needs (see Section 4). Agents could also check individually (against the protocol specification) if they have actions that *match* with the constitutive rules independently from the regulative specification.

Baldoni et al. [2013] analyze alternative proposals for the specification of interaction protocols based on commitments, showing that these proposals still miss the desired clear distinction. None allows the specification of both parts (1) *in a decoupled way*, (2) *by means of first-class languages*, (3) *which allow flexible representations*. It, then, proposes a model that extends commitment-based interaction protocols with an explicit regulative specification, given as a set of *constraints among commitments*. For representing the latter, language 2CL is proposed. The language is characterized by a graphical notation, aimed at supporting the designer in the specification of interaction protocols.

2 Constitutive and Regulative Specifications

A commitment $\mathsf{C}(x, y, r, p)$ denotes that agent x commits to an agent y to bring about the consequent condition p when the antecedent condition r holds. When r equals $true$, the short notation $\mathsf{C}(x, y, p)$ is used. Agents share a social state that contains commitments and other literals that are relevant to their interaction. Agents affect the social state by executing actions, defined in terms of operations onto the social state (e.g. adding a new commitment, releasing another agent from some commitment [Yolum and Singh, 2002]). On the other hand, agent behavior is affected by commitments because agents are expected to respect the commitments they have taken. Thus, commitments have a *regulative* nature.

A commitment protocol [Yolum and Singh, 2001; 2002; Chopra, 2009] is a set of actions, whose semantics is known to (and agreed upon by) all of the participants. Most works on agents adopt a *precondition-effect* view of actions. *Preconditions* can either be to action execution or to some effect. The former are literals that must hold in the social state to make the action executable, the latter are additional conditions that enable the production of the specific effect that they control. For instance, in order to pay by credit card it is necessary to own a credit card (precondition to the action). If a credit card owner uses it for paying, the payment will be done only if the card is valid (conditional effect). Protocol actions are specified by means of *constitutive rules*.

In many applications, it is often necessary to specify *patterns of interaction* which require a degree of expressiveness that, as we will show, commitments alone do not have. Contrarily to constitutive rules, which define new forms of behavior, these patterns regulate antecedently existing forms of behavior [Searle, 1995]: such rules regulate the social reality, defined by the constitutive rules. For this reason we refer to them as the *regulative specification of the protocol*. For example, a purchase protocol may state that the payment must occur first in order for the shipment to proceed. A regulative specification may, therefore, be viewed as encoding a *policy*.

The current proposals show some limits in the realization of this model, each with its pros and cons. Fornara and Colombetti's proposal [2004a; 2004b] is too rigid: the use of interaction diagrams conflicts with the desirable flexibility of commitments because it forces the ordering of action executions. In this respect, ConDec's use of constraints [Pesic and van der Aalst, 2006] is better: the declarative approach that is proposed (which does not involve commitments nor is set in an agents' framework) is aligned with the declarative nature of commitments. The problem is that constraints are used with a constitutive and not with a regulative aim. In other words, the ConDec approach uses the constraints to define a process with its execution traces. Moreover, they are defined in terms of performing actions rather than of bringing about conditions. Chopra and Singh [2008] propose an implementation where the regulative specification is given on top of actions: while commitments are given on conditions and not on the actions that should bring them about, constraints are posed on the action execution, with the result that modularity is not obtained. So, in order to impose that sending goods should follow payment, the action *send-goods* should have as a precondition a literal that is made true as an effect of the action *pay*. This solution has two limits. The first is that it does not abide by the meaning of "regulative" rule, i.e. to regulate antecedently existing forms of behavior. Preconditions are, in fact, part of the constitutive specification of the actions because they help defining the rules of the game and to specify the possibility of action [Searle, 1995; La Torre, 2010; Grossi, 2007]. Moreover, their use forces the *regimentation* of the regulation [Jones and Sergot, 1994] because it is not possible to execute actions leading to violations. The adoption of regimentation rather than enforcement should, however, be left up to the designer of the system and not be imposed by the specification framework [Grossi *et al.*, 2007]. The same holds for [Winikoff *et al.*, 2005; Yolum and Singh, 2002; 2001; Singh, 2003; El-Menshawy *et al.*, 2010; 2011].

Our proposal (Figure 1) aims at overcoming the listed limits.

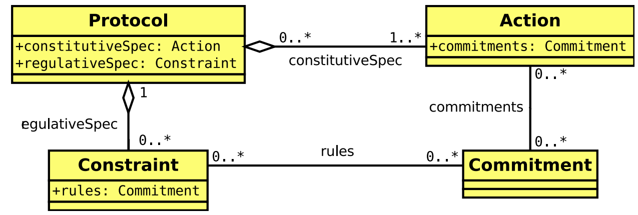

Figure 1: Decoupling between constitutive (actions) and regulative (constraints) specifications.

We extend commitment-based protocols by adding the possibility of defining expressive regulative specifications, that are not limited to commitments but that account for patterns of interaction. To this aim we propose the use of a declarative language, 2CL, which allows the definition of constraints on the evolution of the social state. Such specifications preserve the flexibility of commitment protocols because they do not force agents to execute given paths but rather let them free to choose their courses of actions, as long as they respect the rules. 2CL constraints relate commitments and not actions/events. This modularity facilitates the design of interaction protocols because it allows for a separate specification of the actions and of the regulative part, once the vocabulary of terms (facts, commitments) that can appear in the social state is defined. It also facilitates the re-use of previously defined actions or regulations as long as the domain of discourse does not change. Regulations, in fact, are not hidden inside actions, and actions are not over-specified for the need of including regulations in their preconditions. This is, for instance, useful in the case when a designer must tailor an interaction protocol to particular needs, a case that is discussed in the paper. Another characteristic of the proposed language is that it is a general tool, that leaves the designer free to choose how to implement the specified regulations (e.g. by enforcement or by regimentation) rather than forcing him/her in any direction.

3 2CL Commitment Protocols

We extend commitment-based protocols from [Chopra and Singh, 2008; Chopra, 2009] by adding a *regulative* specification, which captures the legal evolutions of the social state. An interaction protocol *P* is a tuple $\langle Ro, F, A, C \rangle$, where Ro is a set of roles, identifying the interacting parties, F is a set of literals (including commitments) that can occur in the social state, A is a set of actions, and C is a set of constraints. The set of social actions A, defined on F and on Ro, forms the *constitutive specification* of the protocol, while the set of constraints C, defined on F and on Ro again, forms the *regulative specification* of the protocol.

We use as a running example the well-known FIPA Contract Net Protocol (CNP for short), that includes two roles, the initiator (i) and the participant (p). The initiator calls for proposals. The participant may send a proposal or refuse to do it. When a proposal is received, the initiator may either reject or accept it. We do not model the exchange of information concerning the proposal itself but only the interaction concerning the task assignment and solution.

We adopt the *constitutive specification* of an action in [Chopra, 2009]. An action meaning is given in terms of how it affects the social state by adding or removing literals or by performing operations on the commitments (create, release, etc., see [Singh, 1999; Yolum and Singh, 2001]). The constitutive specification follows the grammar below, where the *means* construct amounts to a *counts-as* relation [Searle, 1995]:

$A \rightarrow (Action$ **means** $Operation)^+$
$Action \rightarrow protocolAction([paramList])$
$Operation \rightarrow Op(commitment) \mid fact \mid$
$\quad Operation \wedge Operation$
$Op \rightarrow$ CREATE | DELETE | RELEASE | DELEGATE | ...

protocolAction identifies an interactive (observable) action of the protocol; *paramList* denotes its parameters; *Op* is a commitment operation; *commitment* is a commitment of form $\mathsf{C}(x, y, r, p)$ (see also [Chopra, 2009, page 49]), where x and y are roles in Ro and r and p are formulas in disjunctive normal form of propositional literals in F; and *fact* is a positive or negative proposition that does not concern commitments and which contributes to the social state.

These are the CNP actions as expressed by the grammar:

(a) $send\_cfp(i, p)$ **means** CREATE($\mathsf{C}(i, p, assigned\_task(i, p))$)
(b) $send\_proposal(p, i)$ **means** CREATE($\mathsf{C}(p, i, solved\_task(p, i))$)
(c) $send\_refusal(p, i)$ **means** $refused\_task(p, i) \wedge$
RELEASE($\mathsf{C}(i, p, assigned\_task(i, p))$)
(d) $send\_accept(i, p)$ **means** $assigned\_task(i, p)$
(e) $send\_reject(i, p)$ **means** $rejected\_proposal(i, p) \wedge$
DELETE($\mathsf{C}(i, p, assigned\_task(i, p))$)$\wedge$
RELEASE($\mathsf{C}(p, i, solved\_task(p, i))$)
(f) $send\_done(p, i)$ **means** $solved\_task(p, i)$
(g) $send\_failure(p, i)$ **means** $failed(p, i) \wedge$
DELETE($\mathsf{C}(p, i, solved\_task(p, i))$)

They are quite straightforward. For instance, *send_cfp* creates $\mathsf{C}(i, p, assigned\_task(i, p))$, which states the resolution of the initiator to assign a task to a participant. During the execution the participant may refuse to solve the task or the initiator may reject a proposal, e.g. because not convenient.

To represent *regulative specifications*, we propose the declarative language 2CL (Constraints among Commitments Language). 2CL has also a graphical notation, inspired by ConDec [Pesic and van der Aalst, 2006] and by [Baldoni *et al.*, 2011a]. By means of it, the designer can express many kinds of constraints describing the legal evolutions of the social state. As underlined in [Pesic and van der Aalst, 2006; Baldoni *et al.*, 2011a; Montali, 2010], constraint-based declarative representations provide abstractions which allow to explicitly capture what is mandatory and what is forbidden, without the need to express the set of possible executions extensionally. For this reason, models remain compact improving flexibility. This is an advantage with respect to procedural approaches, characterized by a prescriptive nature which requires the specification of *all* the allowed evolutions. It also accommodates naturally to the commitment-based approach, where a central issue is the respect of the agents' autonomy. 2CL specifications follow the grammar:

$C \rightarrow (Disj\ op\ Disj)^*$
$Disj \rightarrow Conj$ OR $Disj \mid Conj$ XOR $Disj \mid Conj$
$Conj \rightarrow literal$ AND $Conj \mid literal$

C is a set of constraints of form $A\ op\ B$, where A and B are formulas of literals and *op* is one of the operators described in [Baldoni *et al.*, 2013]; *literal* can be either a commitment or a positive or negative proposition (negation means that the proposition does not hold in the social state). The proposed operators capture those kinds of constraints that the analysis of protocols like CNP, Net Bill [Baldoni *et al.*, 2011b] and Robert's Rules of Order [Baldoni *et al.*, 2010] suggest being the most useful for the regulative specification of interaction protocols. For each relation, there are two types of constraint: *base* constraints express what should become true in the social state and when; *persistence* constraints capture conditions that should hold in all the traversed states, until another condition of interest becomes true.

For lack of space, we omit the description of the operators that can be found in [Baldoni *et al.*, 2013], and just report and comment the regulative specification for CNP.

c1: $\mathsf{C}(i, p, assigned\_task(i, p))$ •—▷•
$\mathsf{C}(p, i, solved\_task(p, i))$ XOR $refused\_task(p, i)$
c2: $\mathsf{C}(p, i, solved\_task(p, i))$ •—▷•
$rejected\_proposal(i, p)$ XOR $assigned\_task(i, p)$
c3: $assigned\_task(i, p)$ •—▷• $solved\_task(p, i)$ XOR $failed(p, i)$

The arrow •—▷• represents the cause operator, expressing the constraint that when the antecedent condition holds, afterwards, the consequent condition must become true sooner or later. Accordingly, constraint $c1$ represents that when the initiator must assign a task, the participant is expected to either refuse the task or to take the commitment to solve it, although this does not have to happen immediately at the next execution step. The specification foresees that the participant cannot take the initiative of proposing to solve a task (or of refusing to do something) if the initiator has not declared that there is a task to solve. Notice that we have not mentioned which actions should be executed to change the social state: any action, whose effect is compatible with the given schema of evolution of the social state is feasible. $c2$ and $c3$ exploit the same 2CL operator.

4 Tailoring the Contract Net Protocol

We show the versatility of the proposal by showing how easy it is to vary the CNP protocol by playing with its regulative specification separately from the constitutive specification of its actions. The first two simple variants are obtained by changing a single 2CL operator. In both cases the constitutive specifications are left unchanged. If we substitute the *cause* relation in $c1$ with a *before* relation (—▷•), the participant *will not be obliged* to answer with a proposal to the call of the initiator (it is allowed to have an unresponsive behavior); however, when a proposal is made, it must be made only after the call:

c1: $\mathsf{C}(i, p, assigned\_task(i, p))$ —▷•
$\mathsf{C}(p, i, solved\_task(p, i))$ XOR $refused\_task(p, i)$

Instead, if one uses in $c1$ a *response* (•—▷) operator, the participant *can also take the initiative* to solve a task even though the initiator has not made any request (zealous participant), as it happens in the case of advertisement, where a provider promotes its services:

c1: $\mathsf{C}(i, p, assigned\_task(i, p))$ •—▷
$\mathsf{C}(p, i, solved\_task(p, i))$ XOR $refused\_task(p, i)$

Then, we consider a *call for bids* where an initiator publishes an open call, e.g. in an official gazette, that does not require the subscribers to the gazette to answer.

c1: $\mathsf{C}(i, p, assigned\_task(i, p)) \;-\!\!\triangleright\!\!\bullet$
$\qquad \mathsf{C}(p, i, solved\_task(p, i))$ XOR $refused\_task(p, i)$
c2: $\mathsf{C}(p, i, solved\_task(p, i)) \;\bullet\!\!-\!\!\triangleright\!\!\bullet$
$\qquad rejected\_proposal(i, p)$ XOR $assigned\_task(i, p)$
c3: $assigned\_task(i, p) \;\bullet\!\!-\!\!\triangleright\; solved\_task(p, i)$ XOR $failed(p, i)$
c4: $assigned\_task(i, p) \;-\!\!\triangleright\!\!\bullet\; solved\_task(p, i)$

That the participant is not obliged to send a bid is captured by $c1$, which is a *before* ($-\!\!\triangleright\!\!\bullet$) instead of being a *cause* ($\bullet\!\!-\!\!\triangleright\!\!\bullet$). Constraint $c3$ captures that a participant can notify a failure also when the task has not been assigned to it yet if, for some reason, it finds out that it would be impossible for it to proceed with the solution, were the task assigned to it. To capture this behavior, instead of using a *cause* constraint, we use the softer *response* constraint ($\bullet\!\!-\!\!\triangleright$). We also add constraint $c4$ (of kind *before*) to capture that a failure can be notified at any moment while a solution can be communicated only after a task is assigned to the participant.

The last variant leaves a much greater freedom of behavior to the initiator and to the participant.

c1: $\mathsf{C}(p, i, solved\_task(p, i)) \;-\!\!\triangleright\!\!\bullet\; assigned\_task(i, p)$
c2: $refused\_task(p, i) \;\bullet\!\!\not-\!\!\bullet\; \mathsf{C}(p, i, solved\_task(p, i))$
c3: $rejected\_proposal(i, p) \;\bullet\!\!\not-\!\!\bullet\; assigned\_task(i, p)$

The only constraint that is imposed on the evolution of the social state is that a task cannot be assigned to a participant who has not yet committed to solve it ($c1$). Moreover, $c3$ states that proposal assignment and rejection are mutually exclusive ($\bullet\!\!\not-\!\!\bullet$), and $c2$ that the refusal of a task is mutually exclusive to the commitment to solve it. So a participant can express its intention to solve a task for which no call was made, and it can give a solution before it was assigned the task. The initiator can ignore a participant even though it committed to solve the task and not answer to it. It can call for proposals even if it already has a commitment by the participant, and it can reject a participant who did not make any proposal.

5 Discussion

Constraints cannot be represented (and, then, substituted) by proper commitments because their expressive powers are different. For instance, CNP needs to express that the assignment of a task to a participant is to be done prior to the execution of that task. One may think to represent this temporal relationship by a conditional commitment of the kind: $\mathsf{C}(p,$ $i,$ $assigned\_task(i, p),$ $solved\_task(p, i)\ xor\ failed(p, i))$ but commitments do not impose that the antecedent condition be achieved first. If p solved the task before it was assigned to it the commitment is discharged. Instead, $assigned\_task(i, p)$ $\bullet\!\!-\!\!\triangleright\!\!\bullet\; solved\_task(p, i)$ XOR $failed(p, i)$ means that the two conditions are to be achieved in the order.

Another important feature is the *nature of the two engagements*. Ever since their introduction, commitments have been given an explicit normative nature [Castelfranchi, 1995; Singh, 1999]. Also constraints, which define the patterns of interaction of a protocol, have a normative nature, in the sense that they represent what must hold in an execution for a protocol to be respected. Both constraints and commitments, due to their regulative nature, introduce a notion of *violation* but while a commitment condition is to be achieved and, so, commitment violations can be detected only at the end of the interaction, violations to constraints can be detected during the interaction. A constraint is, in fact, like a boundary that should not be crossed.

The classical commitment-based approach is very respectful of the agents' autonomy. Autonomy implies that each agent decides what is the best for itself. So, protocols do not dictate agents when to execute specific actions. This spirit is respected by 2CL, even though our protocols include a regulative specification which restricts the acceptable executions, because 2CL does not regiment the regulative rules. Let us explain the meaning of opportunity, and the possibility to incur into a violation, with the help of a simple example: suppose that at a summer school the official language everybody should speak is English. Every student at the school has a badge reporting the name and the nationality of the person. If a French attendee meets a colleague whose badge says she is from France, the first student might decide to speak in French even though the official language is English. The clear expected advantage is a better understanding. However, the violation of the rules introduces a *risk*: if the second student by mistake took the badge of her roommate, she might not understand the former because the protocol is not attended.

Finally let us consider constraints between commitments with different debtors, like $\mathsf{C}(c,\ m,\ purchase(goods)) \;\bullet\!\!-\!\!\triangleright\!\!\bullet$ $\mathsf{C}(m,\ c,\ sold(goods, price))$ by which the commitment of the customer c to buy some goods is to be followed by the commitment of the merchant m to sell the goods at the agreed price. It may seem, in this case, that the autonomy of the merchant is reduced by the fact that another agent took some commitment, but it is not so. In fact, since constraints are supposed to be public and inspectable specifications, an agent, willing to play a role in a protocol, has the means for understanding if that pattern of interaction meets its goals. By autonomously deciding whether entering the protocol it, however, *commits* to respect its rules along the *whole* interaction. The fact that all agents accept to respect the rules has the advantage of making the course of interaction predictable and, therefore, of giving guarantees to all of the participants.

[Baldoni *et al.*, 2014] introduces a software engineering methodology (2CL Methodology) for designing 2CL protocols, for specialising them, and for composing a new 2CL protocols based on a set of given 2CL protocols. It also presents a set of integrated software tools for the design and the analysis of 2CL protocols. Moreover, [Marengo *et al.*, 2011] introduces temporal expressions as antecedent and consequent conditions of commitments and formalizes commitment progression along the occurrence of events.

References

[Baldoni *et al.*, 2010] Matteo Baldoni, Cristina Baroglio, and Elisa Marengo. Behavior-Oriented Commitment-Based Protocols. In *Proc. of ECAI*, volume 215 of *Frontiers in AI and Appl.*, pages 137–142. IOS Press, 2010.

[Baldoni *et al.*, 2011a] Matteo Baldoni, Cristina Baroglio, Ingo Brunkhorst, Nicola Henze, Elisa Marengo, and Viviana Patti. Constraint Modeling for Curriculum Planning and Validation. *International Journal of Interactive Learning Environments*, 19(1):83–123, March 2011.

[Baldoni *et al.*, 2011b] Matteo Baldoni, Cristina Baroglio, and Elisa Marengo. Commitment-based Protocols with Behavioral Rules and Correctness Properties of MAS. In *Post-Proc. of DALT 2010*, volume 6619 of *LNAI*. Springer, 2011.

[Baldoni *et al.*, 2013] Matteo Baldoni, Cristina Baroglio, Elisa Marengo, and Viviana Patti. Constitutive and Regulative Specifications of Commitment Protocols: a Decoupled Approach. *ACM Transactions on Intelligent Systems and Technology*, 4(2):22:1–22:25, March 2013.

[Baldoni *et al.*, 2014] Matteo Baldoni, Cristina Baroglio, Elisa Marengo, Viviana Patti, and Federico Capuzzimati. Engineering commitment-based business protocols with 2CL methodology. *JAAMAS*, 28(4):519–557, July 2014.

[Castelfranchi, 1995] Cristiano Castelfranchi. Commitments: From Individual Intentions to Groups and Organizations. In *Proc. of ICMAS 1995*, pages 41–48. The MIT Press, June 1995.

[Cherry, 1973] Christopher Cherry. Regulative Rules and Constitutive Rules. *The Philosophical Quarterly*, 23(93):301–315, 1973.

[Chopra and Singh, 2008] Amit K. Chopra and Munindar P. Singh. Constitutive Interoperability. In *Proc. of AAMAS 2008*, pages 797–804. IFAAMAS, 2008.

[Chopra and Singh, 2009] Amit K. Chopra and Munindar P. Singh. Multiagent Commitment Alignment. In *Proc. of AAMAS 2009*, pages 937–944. IFAAMAS, 2009.

[Chopra, 2009] A.K. Chopra. *Commitment Alignment: Semantics, Patterns, and Decision Procedures for Distributed Computing*. PhD thesis, North Carolina State University, Raleigh, NC, 2009.

[El-Menshawy *et al.*, 2010] Mohamed El-Menshawy, Jamal Bentahar, and Rachida Dssouli. Verifiable Semantic Model for Agent Interactions Using Social Commitments. In *Proc. of LADS 2009*, volume 6039 of *LNCS*, pages 128–152. Springer, 2010.

[El-Menshawy *et al.*, 2011] Mohamed El-Menshawy, Jamal Bentahar, and Rachida Dssouli. Symbolic Model Checking Commitment Protocols using Reduction. In *Proc. of DALT 2010*, volume 6619 of *LNCS*. Springer, 2011.

[Fornara and Colombetti, 2004a] Nicoletta Fornara and Marco Colombetti. A Commitment-Based Approach To Agent Communication. *Applied Artificial Intelligence*, 18(9-10):853–866, 2004.

[Fornara and Colombetti, 2004b] Nicoletta Fornara and Marco Colombetti. Protocol Specification Using a Commitment Based ACL. In *Proc. of ACL 2003*, volume 2922 of *LNCS*, pages 108–127. Springer, 2004.

[Grossi *et al.*, 2007] Davide Grossi, Huib Aldewereld, and Frank Dignum. Ubi lex, ibi poena: Designing norm enforcement in e-institutions. In *Proc. of COIN 2006*, volume 4386 of *LNCS*, pages 101–114. Springer, 2007.

[Grossi, 2007] Davide Grossi. *Designing Invisible Handcuffs, Formal Investigations in Institutions and Organizations for Multi-agent Systems*. PhD thesis, University of Utrecht, 2007.

[Jones and Sergot, 1994] Andrew J. I. Jones and Marek Sergot. *On the Characterization of Law and Computer Systems: the Normative Systems Perspective*, pages 275–307. John Wiley & Sons, Inc., New York, NY, USA, 1994.

[La Torre, 2010] Massimo La Torre. *Law as Institution*, volume 90 of *Law and Philosophy Library*. Springer, 2010.

[Marengo *et al.*, 2011] Elisa Marengo AND Matteo Baldoni AND Cristina Baroglio AND Amit K. Chopra AND Viviana Patti AND Munindar P. Singh. Commitments with Regulations: Reasoning about Safety and Control in REGULA. In *Proc. of AAMAS*, pages 467–474. IFAAMAS, 2011.

[Montali, 2010] Marco Montali. *Specification and Verification of Declarative Open Interaction Models: a Logic-Based Approach*, volume 56 of *LNBIP*. Springer, 2010.

[Pesic and van der Aalst, 2006] Maja Pesic and Wil M. P. van der Aalst. A Declarative Approach for Flexible Business Processes Management. In *Proc. of BPM 2006*, volume 4103 of *LNCS*, pages 169–180. Springer, 2006.

[Searle, 1995] John R. Searle. *The construction of social reality*. Free Press, New York, 1995.

[Singh, 1999] Munindar P. Singh. An Ontology for Commitments in Multiagent Systems. *Artificial Intelligence and Law*, 7(1):97–113, 1999.

[Singh, 2000] Munindar P. Singh. A Social Semantics for Agent Communication Languages. In *Issues in Agent Communication*, volume 1916 of *LNCS*, pages 31–45. Springer, 2000.

[Singh, 2003] Munindar P. Singh. Distributed Enactment of Multiagent Workflows: Temporal Logic for Web Service Composition. In *Proc. of AAMAS 2003*, pages 907–914. ACM, 2003.

[Winikoff *et al.*, 2005] Michael Winikoff, Wei Lei, and James Harland. Enhancing Commitment Machines. In *Proc. of DALT 2004*, volume 3476 of *LNCS*, pages 198–220. Springer, 2005.

[Yolum and Singh, 2001] Pinar Yolum and Munindar P. Singh. Designing and Executing Protocols Using the Event Calculus. In *Proc. of AGENTS 2001*, pages 27–28. ACM, 2001.

[Yolum and Singh, 2002] Pinar Yolum and Munindar P. Singh. Commitment Machines. In J.-J. Ch. Meyer and M. Tambe, editors, *Proc. of ATAL 2001*, volume 2333 of *LNCS*, pages 235–247, Seattle, WA, USA, August 2002. Springer.