Behavior-Oriented Commitment-based Protocols

Matteo Baldoni, Cristina Baroglio, and Elisa Marengo[1]

Abstract. Ever since the seminal work of Searle, two components of interaction protocols have been identified: constitutive rules, defining the meaning of actions and regulative rules, defining the flow of execution, i.e. the behavior the agent should show. The two parts together define the meaning of the interaction. Commitment-based protocols, however, usually do not account for the latter and, when they do it, they do not adopt a decoupled representation of the two parts. A clear distinction in the two representations would, however, bring many advantages, mainly residing in a greater openess of multi-agent systems, an easier re-use of protocols and of action definitions, and a finer specification of protocol properties. In this work we introduce the notion of behavior-oriented commitment-based protocols, which account both for the *constitutive* and the *regulative* specifications and that explicitly foresee a representation of the latter based on *constraints among commitments*. A language, named 2CL, for writing regulative specifications is also given.

1 Introduction

An interaction protocol is a pattern of behavior that allows a set of agents to engage expected cooperations with one another, when playing its roles. One of the most successful approaches to the specification of interaction protocols is represented by *commitment-based protocols*, introduced by Singh and colleagues [11, 30, 38, 37]. In this context a commitment is a literal, that can hold in the social state of the system, representing the fact that a debtor committed to a creditor to bring about some condition. In order to understand each other and cooperate, the agents that play the protocol roles share the semantics of a set of social actions, whose execution affects the social state by creating new commitments, canceling commitments, and so forth. The only constraint that commitment protocols include, to say that an interaction is successful, is that all commitments are discharged.

These characteristics give commitment-based protocols great flexibility and give to the involved agents great autonomy. In fact, they are free to apply the social actions in any order they wish if, in the end, commitments are discharged. However, in our opinion, such protocols allow too much freedom in some practical contexts. Let us see an example. Let us consider the action *shaking hands*, which means "agreement reached" in a negotiation protocol. A commitment-based protocol would specify the meaning of shaking hands but it would not put any constraint on when it makes sense to use the action. What if a person shakes hands with someone he/she would like to reach an agreement with *before* starting the negotiation? Executing the action in that context makes no sense and may lead to misunderstandings. Something seems to be missing in the specification. Indeed, as Cherry observes [7] when commenting Searle's work [28], actions acquire *additional* meaning depending on the *context* where they are used: in the example, the context is given by a particular point in the evolution of the interaction between the parties. Sometimes for filling this gap commitment-based protocols enrich actions with *preconditions* to their (non-) executability, e.g. [35, 12, 18]. Preconditions capture the specific kind of context in which actions *can* be used; by means of them it becomes possible to rule the order of action execution. However, putting the context inside preconditions brings to an *over-specification* of actions. If the context were, in some way, accounted for separately, by rules concerning the possible evolutions of the content of the social state, actions would be simpler and easier to understand because their specification would correspond to the definition of the action *per se* and not of the action in a context of reference. Moreover, it would be possible to represent a wider range of evolution-ruling relations, without imposing the mere sequencing.

In our view, an interaction protocol must not only specify the agreed meaning of actions but it must express also an agreement on the way the agents will behave and use the protocol actions. This should be done in a way that does not compromise the autonomy of agents, which would be free to decide how to act and to take advantage of opportunities, that arise along the interaction, taking also the risk of being misunderstood when they get out of the boundaries given by the protocol. This should be done also in a way that preserves the flexibility of the protocol. After an agreement we can shake hands twice, if we are happy to do so, but shaking hands before the agreement is not understandable in the context of that protocol.

In this work, we face this issue starting from Chopra and Singh's distinction between the *constitutive* and *regulative* specifications of the interaction [13], deriving from the work of Searle [28], and propose a new model of commitment-based interaction protocol. The main characteristic of this model is a decoupled representation of the constitutive and the regulative specifications of the protocol, which are both based on commitments. While the *constitutive* specification defines the meaning of actions based on their effects on the social state, the *regulative* specification is a set of behavioral rules, given in terms of *constraints among commitments* (and literals), which regulate the evolution of the social state *independently from* the executed actions. To the best of our knowledge, such a sharp decoupling, was not implemented in commitment-based interaction protocols before. We use, as a running example, the *Robert's Rules of Order* [25] (RONR : Robert's Rules of Order New Revision [26]), a well-known regulation of the behavior to be followed by a democratic deliberative assembly, like Parliament, in order to discuss and take decisions about issues called *motions*.

2 Constitutive and Regulative Specifications

Commitment protocols [30, 37, 38] are interaction patterns given in terms of commitments, involving a set of predefined roles. Commitments are directed from a debtor to a creditor. The notation

[1] Dip. di Informatica, Univ. di Torino, Italy, email: {baldoni,baroglio,emarengo}@di.unito.it

$C(x, y, r, p)$ denotes that the agent playing the role x commits to an agent playing the role y to bring about the condition p when the condition r holds. All commitments are conditional. An unconditional commitment is merely a special case where r equals $true$. Whenever this is the case, we use the short notation $C(x, y, p)$. Agents share a social state that contains commitments and other literals that are relevant to their interaction. Every agent can affect the social state by executing actions, whose definition is given in terms of updates to the social state (e.g. add a new commitment, release another agent from some commitment, satisfy a commitment). So a commitment protocol is made of a *set of actions*, involving the foreseen roles and whose semantics is agreed upon by all of the participants [37, 38, 10].

On the other hand, an autonomous agent situated in an environment *decides* which actions to perform depending on the particular situation it is facing. Agents show a *behavior*, which is not captured by the action definitions but rather it involves a decision process (e.g. a procedure or a goal-driven plan [34]) aimed at selecting the action to execute [36, 27]. As an example, let us consider the *Robert's Rules of Order* (RONR) [25, 26], which is one of the best known parliamentary laws for ruling democratic and deliberative assemblies. RONR foresees two roles: the *chair* of the assembly and the *participants* to the assembly. The activity of the assembly basically consists in discussing a motion at a time, and then voting. The rules are aimed at guaranteeing that the assembly works in a democratic way. Among other rules, in particular, it specifies that voting will not take place until all the participants, who raised their hand for expressing their opinion, have spoken; it is not allowed to different members to speak at the same time; and in order to speak one must have the floor. As long as everybody *behaves* according to the rules, the assembly works in a democratic way. In other terms, RONR not only specifies the actions to use but it rules the behavior of the participants and of the chair (specifying the contexts in which the execution of actions makes sense) so to guarantee the success of the assembly. The participants autonomously decide whether conforming to the rules. As long as they do it, they are sure *to have certain rights*.

The same dichotomy between actions and behavior was pinned out by Searle [28] and other authors, e.g. [7, 6, 13], who proposed a distinction between the *regulative* and the *constitutive* specifications of interaction protocols: the latter gives the semantics of actions, while the former rules the flow of execution. The adoption of a representation that includes both these parts is fundamental in all those contexts where the protocol itself includes actions and constraints on the behavior, as in RONR. If we removed the constraints, the agents would gain a great flexibility but at the cost of losing certain guarantees. What is more, we claim that the two specifications should be as separated as possible, and that it should be possible to modify the one without the need of modifying the other. The advantages of the decoupling are an easier *re-use* of actions in different contexts, a simpler *customization* and *composition* of protocols.

In the literature there are many proposals based on regulative and/or constitutive specifications, however, as we discuss in details in Section 4, they all show some limits in the realization of a decoupled complete model, that we overcome with our proposal. These limits affect the *openness*, *interoperability*, and *modularity* of design of multi-agent systems. Some proposals lack the decoupling between the regulative and constitutive specifications because either they include the regulative specification in the *action* definition (i.e. they mix the regulative and constitutive parts) [38, 35, 11, 18, 13] or they specify regulative rules by means of constraints among *actions* [31, 2, 23, 9, 20]. Other proposals lack either the constitutive part [24] or the regulative part [37, 10]. Finally, others adopt too rigid models (e.g. [16, 17] use interaction diagrams) to specify the desired behavior. This conflicts with the flexibility of commitments.

To overcome all these limits, we propose the use of a declarative language, named 2CL, for capturing constraints that rule the execution flow. The inspiration is from [24] and is adapted to the regulation of agent interaction protocols instead of business processes. The differences are that our proposal includes a constitutive specification of actions (which misses in [24]), and that constraints *relate commitments* (more in general, literals) and *not actions*. Doing so we can capture also the additional meaning of an action given by the context in which it is used. Moreover, we are able to endorse the "guarantees" foreseen by a protocol to the participants. Our solution allows the achievement of all these aspects in a modular way, maintaining the same flexibility of commitment-based protocols and allowing to gain an easier re-use of actions in different contexts and an easier re-use of protocols with different actors. Protocols can be modified more easily, allowing greater openness and a better support to the verification of properties.

3 Behavior-oriented Commitment-based Protocols

In this section we propose a representation of commitment-based protocols which encompasses a *constitutive* specification, defining the meaning of actions for all the agents in the system, and a *regulative* specification, constraining the possible evolutions of the social state.

Definition 1 (Interaction protocol) *An interaction protocol* P *is a tuple* $\langle R, F, A, C\rangle$*, where* R *is a finite set of roles, identifying the interacting parties,* F *is a finite set of literals (including commitments) that can occur in the social state,* A *is a finite set of actions, and* C *is a finite set of constraints.*

The set of social actions A, defined on F and on R, forms the *constitutive specification* of the protocol, while the set of constraints C, defined on F and on R, forms the *regulative specification* of the protocol. Each role is identified by a unique label. F is a set of literals. Each literal can be a commitment or a positive or negative proposition that does not concern commitments and that contributes to the social state (they are the conditions that are brought about). The set F represents the domain model and defines the vocabulary used by all agents (through roles) to communicate in the context of the protocol. **Constitutive Specification** defines the meaning of actions in the very same way as it is done in [10], i.e. in terms of how it affects the social state by adding or removing literals or by performing operations on the commitments, like create, delete, discharge (see [29, 38]).

Example 2 (Constitutive Specification of RONR) *The constitutive specification of RONR is:*

(a) $motion(c, m)$ ***means*** $\forall_{p_i \in P}\text{CREATE}(C(c, p_i, cfv(m)))$
(b) $openDebate(c, p, m)$ ***means*** $\text{CREATE}(C(c, p, assignFloor(p, m)))$
(c) $refuseFloor(p, c, m)$ ***means*** $refusedFloor(p, m) \wedge$ $\text{RELEASE}(C(c, p, assignFloor(p, m)))$
(d) $askFloor(p, c, m)$ ***means*** $\text{CREATE}(C(p, c, discussed(p, m)))$
(e) $recognition(c, p, m)$ ***means*** $assignFloor(p, m)$
(f) $startTalk(p, m)$ ***means*** $discussing(p, m)$
(g) $stopTalk(p, m)$ ***means*** $discussed(p, m) \wedge \neg discussing(p, m) \wedge$ $\neg assignFloor(p, m)$
(h) $timeOut(c, p)$ ***means*** $discussed(p, m) \wedge \neg discussing(p, m) \wedge$ $\text{RELEASE}(C(p, c, discussed(p, m))) \wedge \neg assignFloor(p, m)$
(i) $cfv(c, m)$ ***means*** $cfv(m)$
(l) $vote(p, m)$ ***means*** $voted(p, m)$

*To open a motion, the chair creates a commitment for each participant (*P *is the set of all members of the assembly), by which the* *chair commits to let the participant vote the motion, (a). These commitments will be discharged when the chair calls for votes, (i). These two actions are meant to guarantee that every participant will have the possibility to vote the motion. The other actions are used in the interaction of the chair with every single participant. For instance, the action* askFloor *is performed by one of the participants, who commits to discuss the motion, (d). This commitment is discharged by* stopTalk*, (g), which means that the participant has finished to speak and, for this reason, it loses the floor. The other actions are quite straightforward.*

An agent willing to play a role in a protocol, must understand and accept the meaning of the social actions, which is the same for all agents. In order to play the role, the agent must associate to the social actions it should perform one or more of its own actions by means of a *count-as* relation [10].

Regulative Specification is defined by *constraint-based representation*. Due to the declarative nature of the specification, *any* evolution that respects the relations involving the specified literals (including commitments) is allowed. Notice that constraints *do not* specify *which* actions should bring conditions about. This allows the *decoupling* between the constitutive and the regulative specifications. The regulative specification follows the grammar:

$$C \rightarrow (Disj\ op\ Disj)^{+}$$
$$Disj \rightarrow Conj \vee Disj \mid Conj$$
$$Conj \rightarrow literal \wedge Conj \mid literal$$

C is a set of constraints of the form $A\ op\ B$, where A and B are formulas of literals in disjunctive normal form (DNF) and *op* is one of the operators in Table 1; *literal* can either be a commitment or a positive or negative proposition (where negation means that a certain literal does not hold in the social state). Such constraints rule the evolution of the social state by imposing specific patterns on how social states can progress.

The names of the operators and in the graphical format, used in Figure 1, are inspired by ConDec [24]. The semantics of the operators is given in *linear temporal logic* (LTL, [15]), which includes temporal operators such as next-time ($\bigcirc$), eventually ($\Diamond$), always ($\Box$), weak until ($\cup$). Let us describe the various operators. For simplicity the descriptions are given on single literals rather than DNF formulas. For each relation, there are two types of constraint: *base* and *persistence*. Constraints of type *base* express relations between the literals, saying what and when should become true in the social state. Constraints of type *persistence* expresses the fact that a condition of interest holds in all the states until another condition of interest becomes true.

Correlation: (*Base*) whenever a occurs, also b occurs but there is no temporal relation between the two. Negation: if a occurs in some execution, b must not occur. (*Persistence*) whenever a occurs, b must occur in the same state. Negation: when a occurs, b cannot occur in the same state.

Co-existence: the mutual correlation between a and b. Its negation captures the mutual exclusion of a and b.

Response: (*Base*) if a occurs b must hold at least once afterwards (or in the same state). It does not matter if b already held before a. Negation: if a holds, b cannot hold in the same state or after. (*Persistence*) as a difference, a persists until b become true. Negation: if a occurs it does not persist until b.

Before: (*Base*) b does not hold until a becomes true. Afterwards, it is not necessary that b becomes true. Negation: in case b becomes true, a cannot hold beforehand. (*Persistence*) b does not hold until a becomes true and afterwards a holds until b becomes true. Negation: same as negation of base case.

Cause: conjunction of the *response* and *before* relations. This relation captures a form of causality between the antecedent and the consequent [19], i.e. order matters.

Premise: (*Base*) it concerns *subsequent* states: a must hold in all the states immediately preceding a state in which b holds. Negation: a must never hold in a state that immediately precedes one where b holds.

Immediately After: (*Base*) it concerns *subsequent* states: b must occur in all the states immediately following a state where a occurs. Negation: b does not hold in the states immediately following a state where b holds.

Proposition 3 *From the definitions in Table 1, every positive persistent constraint relation implies the corresponding positive basic constraint relation and every negative basic constraint relation implies the corresponding negative persistent constraint relation.*

Example 4 (Regulative Specification of RONR) *This protocol is aimed at guaranteeing that a motion be discussed and voted in a democratic way. Briefly, it states that before voting everybody who wishes to speak must have the possibility of doing it, in order to speak it is necessary to have the floor, it is not possible to speak while someone else is speaking. However, no limit is imposed on the duration of the discussion. It is up to the chair to decide how many times and to whom assigning the floor. The regulative specification of RONR can be specified in* 2CL *as follows:*

c1: $C(c,p,cfv(m)) \bullet\!\!-\!\!\rhd\!\!\bullet\ C(c,p,assignFloor(p,m))$
c2: $C(c,p,assignFloor(p,m)) \bullet\!\!-\!\!\rhd\!\!\bullet$
$\quad C(p,c,discuss(p,m))$ xor $refusedFloor(p,m)$
c3: $C(p,c,discuss(p,m)) \bullet\!\!-\!\!\rhd\!\!\bullet\ assignFloor(p,m)$
c4: $assignFloor(p,m) \bullet\!\!\Rightarrow discussed(p,m)$
c5: $assignFloor(p,m) \rhd\!\!\rhd\!\!-\ discussing(p,m)$
c6: $discussing(p,m) \bullet\!\!-\!\!\rhd\ discussed(p,m)$
c7: $discussed(p,m) \bullet\!\!-\!\!\rhd\!\!\bullet\ \neg assignFloor(p,m)$
c8: $refusedFloor(p,m) \vee discussed(p,m) \bullet\!\!-\!\!\rhd\!\!\bullet\ cfv(m)$
c9: $cfv(m) -\!\!\rhd\!\!\bullet\ voted(p,m)$
c10: $assignFloor(X,\_) \bullet\!\!\not\!\!-\!\!\bullet\ assignFloor(Y,\_) \wedge X \neq Y$

c1 ($\bullet\!\!-\!\!\rhd\!\!\bullet$) states that if the chair has committed to allow a participant to vote on a motion, then it commits to give the participant the floor; *c2* states that if the social state contains a commitment to assign the floor to a participant, the participant can alternatively commit to discuss or refuse to speak; *c3* states that if a participant committed to discuss a motion, sooner or later it will be given the floor; *c4* ($\bullet\!\!\Rightarrow$) states that when the floor is given to a participant, that participant will keep it until it will finish to speak; *c5* ($\rhd\!\!\rhd\!\!-$) states that the assignment of the floor is a premise to the discussion; *c6* ($\bullet\!\!-\!\!\rhd$) states that if a participant starts to discuss a motion, sooner or later it has to finish; *c7* ($\bullet\!\!-\!\!\rhd$) states that once a participant has finished to speak, then it will lose the floor; *c8* states that votes will be called for only after the participant has either refused to speak or has already spoken. Notice that since the action *motion* creates a commitment of kind $C(c,p,cfv(m))$ for every participant, the call for votes can be done only after all of them have either spoken or refused to do it; *c9* ($-\!\!\rhd\!\!\bullet$) states that voting can be done only after the respective call; *c10* ($\bullet\!\!\not\!\!-\!\!\bullet$) states that the floor cannot be assigned to two participants at the same time. Figure 1 reports these constraints as a graph, whose nodes (the rectangles) contain literals that should be in the social state

Relation	*Type*	*Positive*	*LTL meaning*	*Negative*	*LTL meaning*
Correlation	base	$A \bullet\!\!- B$	$\Diamond A \supset \Diamond B$	$A \bullet\!\!\not- B$	$\Diamond A \supset \neg\Diamond B$
	persistence	$A \bullet\!\!= B$	$\Box(A \supset (A \wedge B))$	$A \bullet\!\!\not= B$	$\Box(A \supset \neg(A \wedge B))$
Co-existence	base	$A \bullet\!\!-\!\!\bullet B$	$A \bullet\!\!- B \wedge B \bullet\!\!- A$	$A \bullet\!\!\not-\!\!\bullet B$	$A \bullet\!\!\not- B \wedge B \bullet\!\!\not- A$
	persistence	$A \bullet\!\!=\!\!\bullet B$	$A \bullet\!\!= B \wedge B \bullet\!\!= A$	$A \bullet\!\!\not=\!\!\bullet B$	$A \bullet\!\!\not= B \wedge B \bullet\!\!\not= A$
Response	base	$A \bullet\!\!-\!\!\triangleright B$	$\Box(A \supset \Diamond B)$	$A \bullet\!\!\not-\!\!\triangleright B$	$\Box(A \supset \neg\Diamond B)$
	persistence	$A \bullet\!\!=\!\!\triangleright B$	$\Box(A \supset (\Diamond B \wedge (A \cup B)))$	$A \bullet\!\!\not=\!\!\triangleright B$	$\Box(A \supset \neg(A \wedge B))$
Before	base	$A -\!\!\triangleright\!\!\bullet B$	$\neg B \cup A$	$A \not-\!\!\triangleright\!\!\bullet B$	$\Box(\Diamond B \supset \neg A)$
	persistence	$A =\!\!\triangleright\!\!\bullet B$	$\neg B \cup (A \cup B)$	$A \not=\!\!\triangleright\!\!\bullet B$	$\Box(\Diamond B \supset \neg A)$
Cause	base	$A \bullet\!\!-\!\!\triangleright\!\!\bullet B$	$A \bullet\!\!-\!\!\triangleright B \wedge A -\!\!\triangleright\!\!\bullet B$	$A \bullet\!\!\not-\!\!\triangleright\!\!\bullet B$	$A \bullet\!\!\not-\!\!\triangleright B \wedge A \not-\!\!\triangleright\!\!\bullet B$
	persistence	$A \bullet\!\!=\!\!\triangleright\!\!\bullet B$	$A =\!\!\triangleright\!\!\bullet B \wedge A \bullet\!\!=\!\!\triangleright B$	$A \bullet\!\!\not=\!\!\triangleright\!\!\bullet B$	$A \not=\!\!\triangleright\!\!\bullet B \wedge A \bullet\!\!\not=\!\!\triangleright B$
Premise	base	$A \triangleright\!\!\triangleright\!\!- B$	$\Box(\bigcirc B \supset A)$	$A \triangleright\!\!\triangleright\!\!\not- B$	$\Box(\bigcirc B \supset \neg A)$
Immediately after	base	$A -\!\!\triangleright\!\!\triangleright B$	$\Box(A \supset \bigcirc B)$	$A \not-\!\!\triangleright\!\!\triangleright B$	$\Box(A \supset \bigcirc \neg B)$

Table 1. 2CL constraint relations and their semantics in LTL.

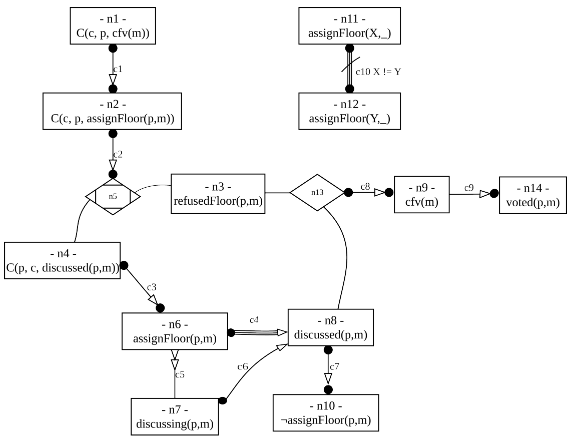

Figure 1. Regulative specification of RONR.

at some point of the execution, while the arrows are operators from Table 1. The diamond represents an "or" of literals/commitments. The bordered diamond represents an "exclusive or". RONR shows how in certain contexts the regulative specification of an interaction protocol is not just a guideline because it is fundamental in order to give guarantees to the participants. On the other hand, the protocol is flexible because, for instance, it does not specify a limited number of rounds of discussion. A participant may also raise the hand as many times as he/she wishes for obtaining the floor. This flexibility is maintained by the 2CL representation. In fact, the constraints do not specify a flow of action execution but rather, following the motto *no flow in flow*, they rule the evolution of the social state in a declarative way, by expressing only what is mandatory and what is forbidden. Moreover, the language allows to easily modify the protocol so to adapt it to different needs. For instance, constraint *c9* does not oblige participants to vote. If one changes the constraint to $\bullet\!\!-\!\!\triangleright\!\!\bullet$ then every participant would be obliged to express its vote.

Notice that the constraints do not specify which actions should be executed to change the social state. *Any* action, whose effects are compatible with the schema of evolution of the social state given by the constraints, is applicable. This respects the spirit of in commitment protocols, which do not specify which action to take in order to satisfy a commitment. The transition from one state to another may even require the application of many actions (not necessarily one). In other words, the regulative specification does *not* give any *procedure* for achieving the social state changes, that it captures: constraints on the evolution of the social state are independent from the actions that are used by the agents. Both, however, are specified on top of the literals in the social state.

Protocols can easily be refined by adding/removing constraints/actions. For instance, if one wants to include in RONR the possibility to postpone a motion (e.g. because it is too late) it is sufficient to allow participants to add the literal *postpone(m)* to the social state in alternative to *refusedFloor(p, m)* and to $C(p, c, discussed(p, m))$ by adding a proper action. Then, to add the constraints: $postpone \bullet\!\!-\!\!\triangleright cfv(postpone(m))$ and $cfv(postpone(m)) -\!\!\triangleright\!\!\bullet voted(p, postpone(m))$, which mean that if a participant proposes to postpone the motion, it is necessary to vote about its postposition.

4 Related works

Singh et al. [37, 10] define interaction protocols in terms of the effects of a set of shared social actions. This approach can be modeled as a special case of our proposal by using an empty regulative specification. This is possible because our proposal enriches the basic commitment protocol model by adding a regulative specification besides the definition of the actions meaning. This is done in a modular way, as hoped for in [32]. In [13] Chopra and Singh introduced the distinction between constitutive and regulative specifications in the definition of commitment-based protocols. In particular, the regulative specification is expressed by *preconditions to the (non-) executability of the actions*. So, for instance in order to impose an ordering between the actions *discuss a motion* and *obtain floor* in the

RONR protocol, the action *discuss a motion* should have as a precondition a literal that is made true as effect of the action *obtain floor*. This solution (adopted also by [18, 38, 35]) is characterized by a *strong localization* of the regulative specification. Both the constitutive and the regulative specifications are indistinguishable in the protocol, being both given by actions. The problem is that by doing so *the definition of an action becomes dependant on the protocol where it is used*. This limits the openness of the system. In our view, a greater decoupling between the actions and the regulative specification would have the advantage of facilitating the re-use of actions because it would allow the avoidance of the over-specification that is necessary to impose an ordering among actions. We can model the proposal by Chopra and Singh [13] as well as those following the same principles by introducing for each action a literal that is univocally associated to it, as an effect of the action, and, then, to define constraints (typically of kind *premise*, $\triangleright\!\!\triangleright\!\!-$) among these literals.

Mallya and Singh [21] propose to order the possible executions according to a set of preferences that take into account the policies of the various parties. No execution is strictly forbidden but a preference criterion is specified. The preference rules are given in terms of *actions*. Preferences do not precisely correspond to regulative rules because they do not constrain the execution flow, nevertheless, giving them in terms of actions makes the specification less flexible and less easily adaptable or open. The same limits can be ascribed to [31] (which inspired [21]), although in this work it is possible to recognize the introduction of a regulative specification, based on the *before* relation applied to events.

Pesic and van der Aalst's [24] proposal, which does not build on commitments nor is set in the agents framework, lacks of a constitutive component but uses the declarative language ConDec for representing business processes. Though not exactly corresponding to interaction protocols, business processes specify the expected behavior of a set of interacting parties by constraining the execution of their tasks. The regulative rules are a first-class element of this representation. They are not local to single actions but rather they are constraints that rule the flow of activity execution. In [23, 9, 20], the authors use this approach to specify interaction protocols and service choreographies. To this aim, they integrate ConDec with SCIFF thus giving an *expectation-based* semantics to actions. However, also these proposals show a too tight connection between the regulative rules and actions because such rules define temporal constraints over actions (events). This, in our opinion, clashes with the openness of multi-agent systems. Let us go back to the RONR protocol: *standing up* to ask for floor must precede the *assignment of floor*. Now, if a participant is not in condition to stand up, the only way he/she has to ask for floor is to *raise a hand*. This action would have the same semantics of *standing-up* in terms of commitments. Now, to allow the participant to use this new action, the regulative specification must be changed by adding a rule, saying that *raising hand* (as well as *standing up*) should occur before *floor assignment*. The need of modifying the regulative specification (even in the case when actions have the same semantics), gives *an undesired rigidity to the protocol*. The problem is that the regulative specification is given in terms of *actions*, so, when changing the actions names we need to change regulative specifications as well. Our proposal overcomes these limits because the regulative specification rules the evolution of the social state and not the execution of actions/events but in case the designer wants to constrain the execution of specific actions, he/she can associate a literal to each action, univocally produced as an action effect.

An approach similar to commitment-based protocols is the one introduced in [2], where *expectation-based protocols* are presented. Expectations concern events expected to happen (or not to happen) and can be associated to time points. Protocols are specified by constraining the times at which events occur. As for [23, 9, 20], the limit is that it works directly on events (i.e. actions); by constraining actions the approach lacks of openness, as above.

Fornara and Colombetti define a commitment-based semantics for the speech acts of agent communication languages, like FIPA, and then use *interaction diagrams* to define agent interaction protocols [16, 17]. In this proposal, the social actions are represented by the speech acts and the constitutive specification is given in terms of commitments. The choice of relying on interaction diagrams is, however, very strong because it forces the ordering of action executions, loosing, in our opinion the flexibility aimed at by the adoption of commitment protocols. In case the designer wishes to specify strict sequences of action executions, as it may happen in [16, 17], our proposal allows to do it in a straightforward way, as explained.

5 Conclusion

This work proposes a commitment-based approach to protocol definition, that is inspired by the work of Singh and colleagues [11, 30, 38, 37, 10, 32], which introduces an explicit representation of both constitutive and regulative specifications in the spirit of [28, 7]. Both specifications are given in a declarative way. The constitutive specification gives the meaning of the social actions, in terms of operations on the social state, as in [10]. The regulative specification is given as a set of constraints on the evolution of the social state expressed in 2CL. The semantics of 2CL is currently grounded on LTL but we mean to study the use of other logics, like CTL* used in [5]. The proposed approach keeps the flexibility of commitment-based protocols, *indirectly* ruling the execution of the actions. The regulative specification is introduced because, in our opinion, the mere constitutive specification of actions is not sufficient, because agents have a behavior and this behavior makes them use actions in specific orders. This order gives actions a supplementary meaning, that must be taken into account in the interaction with the others. Since protocols are supposed to give the shared meaning of actions it is necessary that they account also for meaning given by specific ways of using actions, i.e. by patterns of behavior. By our proposal and by exploiting a declarative language, we have proved that it is possible to express this meaning without losing the flexibility of commitment-based protocols. We do this by putting constraints on the evolution of the social state and not on actions because this allows a greater modularity in the specification, with many advantages. In particular, interoperability is supported in a finer way because it is possible to verify it at the level of actions (*constitutive interoperability*), like [13, 10, 14], as well as at the level of regulation rules (*regulative interoperability*). In general, it is possible for agents to be compatible at the level of actions but not at the level of behavior or the other way around. When the agent's behavioral rules restrict the behavior allowed by the protocol regulative specification, it is necessary to check that these restrictions *do not impose constraints* to the other players. In other words, an agent is allowed to restrict its own behavior but it should not limit the freedom of the other agents, as long as they behave as specified by the protocol. For instance, in RONR a chair must give the floor to all participants who desire to speak. A chair that allows only one participant per motion to speak, restricting the protocol specification, cancels the rights of the participants. This kind of restriction should not be allowed. For a deeper discussion see [4].

The modularity given by a decoupled representation allows designers to easily adapt protocols to different contexts. Moreover, it is possible to check properties that concern a single agent, willing to play a role of the protocol, against the protocol and independently from which other agents will play the other roles. Therefore, if an agent in a system is substituted by another agent, it is not necessary to recheck the whole system from scratch, because certain verifications can be *distributed*.

Singh and Chopra propose to model liveness and safety by using *potential causality* of sends and receives of messages. The two properties are characterized by the compatibility among causal orders of sends and receives [19]. However, one of the key points of commitment protocols is that they allow ruling not only sends and receives but any social action whose meaning is agreed upon. For instance, the agents may agree upon the action *opening a motion* and *start debate*, and expect that they are executed in sequence. It is, therefore, necessary to express a more general notion of causality, which may not be so obvious with actions that are not sends or receives. Moreover, causality may be just one possible relation concerning the ordering of actions. 2CL contains a definition of causality as a relation between literals in the social state. Causality is just one out of many kinds of constraints offered by the language.

For what concerns the adaptation of commitment-based protocols to different contexts of usage, the modular nature of our proposal allows the introduction of two levels of refinement: not only refinement at the constitutive level, e.g. by adding actions as in [12, 32], but also at the regulative level. In this latter case, we exploit the declarative nature of 2CL which allows us to produce stricter sets of constraints just by adding new constraints to those included by the more general protocol. The so obtained refinements can be organized in a taxonomy of protocols.

With respect to [33, 8], our proposal does not handle time explicitly so we cannot yet represent and handle timeouts and also compensation mechanisms. We plan to tackle these issues in future work.

Finally, [22] introduces a formal background that allows a procedural composition of protocols (dialogues). In our view, the adoption of a procedural approach reduces the flexibility that protocols should have. We mean to study, as future work, a methodology that allows the achievement of compositionality in declarative protocols. The intuition [3] is that the decoupling of the regulative and constitutive specifications will facilitate the specification of a methodology.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their comments. This work has partially been funded by Regione Piemonte, ICT4LAW project.

REFERENCES

[1] *Declarative Agent Languages and Technologies VII, 7th Int. Workshop, DALT 2009*, *LNCS* 5948. Springer, 2010.

[2] M. Alberti, D. Daolio, P. Torroni, M. Gavanelli, E. Lamma, and P. Mello, 'Specification and verification of agent interaction protocols in a logic-based system', in *Proc. of SAC*, pp. 72–78. ACM, (2004).

[3] M. Baldoni, C. Baroglio, and E. Marengo, 'Constraints among Commitments: Regulative Specification of Interaction Protocols', in *Proc. of Int. Workshop AC 2010*, pp. 2–18, (2010).

[4] M. Baldoni, C. Baroglio, and E. Marengo, 'Commitment-based Protocols with Behavioral Rules and Correctness Properties of MAS', in *Proc. of Int. Workshop DALT 2010*, pp. 66–83, (2010).

[5] J. Bentahar, J.-J. Ch. Meyer, and W. Wan, 'Model checking communicative agent-based systems', *Knowl.-Based Syst.*, **22**(3), 142–159, (2009).

[6] G. Boella and L. W. N. van der Torre, 'Regulative and constitutive norms in normative multiagent systems', in *Proc. of KR*, pp. 255–266. AAAI Press, (2004).

[7] C. Cherry, 'Regulative rules and constitutive rules', *The Philosophical Quarterly*, **23**(93), 301–315, (1973).

[8] F. Chesani, P. Mello, M. Montali, and P. Torroni, 'Commitment tracking via the reactive event calculus', in *Proc. of IJCAI*, pp. 91–96, (2009).

[9] F. Chesani, P. Mello, M. Montali, and P. Torroni, 'Verifying a-priori the composition of declarative specified services', in *MALLOW*, vol. 494 of *CEUR Workshop Proceedings*, (2009).

[10] A. K. Chopra, *Commitment Alignment: Semantics, Patterns, and Decision Procedures for Distributed Computing*, Ph.D. diss., NCSU, 2009.

[11] A. K. Chopra and M. P. Singh, 'Nonmonotonic Commitment Machines', in *Proc. of ACL*, *LNCS* 2922, pp. 183–200. Springer, (2003).

[12] A. K. Chopra and M. P. Singh, 'Contextualizing commitment protocol', in *Proc. of AAMAS'06*, pp. 1345–1352. ACM, (2006).

[13] A. K. Chopra and M. P. Singh, 'Constitutive interoperability', in *Proc. of AAMAS'08*, pp. 797–804, (2008).

[14] A. K. Chopra and M. P. Singh, 'Multiagent commitment alignment', in *Proc. of AAMAS'09*, pp. 937–944, (2009).

[15] E. A. Emerson, 'Temporal and Modal Logic', in *Handbook of Theoretical Computer Science*, volume B, 997–1072, Elsevier, (1990).

[16] N. Fornara and M. Colombetti, 'Defining interaction protocols using a commitment-based agent communication language', in *Proc. of AAMAS'03*, pp. 520–527, (2003).

[17] N. Fornara and M. Colombetti, 'A Commitment-Based Approach To Agent Communication', *Applied AI*, **18**(9-10), 853–866, (2004).

[18] L. Giordano, A. Martelli, and C. Schwind, 'Specifying and verifying interaction protocols in a temporal action logic', *J. Applied Logic*, **5**(2), 214–234, (2007).

[19] L. Lamport, 'Time, clocks, and the ordering of events in a distributed system', *Communication of the ACM*, 558–565, (1978).

[20] Montali M., M. Pesic, W.M. P. van der Aalst, F. Chesani, P. Mello, and S. Storari, 'Declarative specification and verification of service choreographies', *ACM Transactions on the Web*, (2009).

[21] A. U. Mallya and M. P. Singh, 'Introducing preferences into commitment protocols', in *AC*, *LNCS* 3859, pp. 136–149. Springer, (2006).

[22] P. McBurney and S. Parsons, 'Games That Agents Play: A Formal Framework for Dialogues between Autonomous Agents', *J. of Logic, Language and Information*, **11**(3), 315–334, (2002).

[23] M. Montali, *Specification and Verification of Declarative Open Interaction Models - A Logic-based framework*, Ph.D. dissertation, University of Bologna, 2009.

[24] M. Pesic and W. M. P. van der Aalst, 'A Declarative Approach for Flexible Business Processes Management', in *Proc. of. BPM Workshops*, *LNCS* 4103, pp. 169–180. Springer, (2006).

[25] H. M. Robert, *Robert's Rules of Order*, 1876. Available at http://www.gutenberg.org/etext/9097.

[26] H. M. III Robert, W. J. Evans, D. H. Honemann, and T. J. Balch, *Robert's Rules of Order Newly Revised, 10th Ed.*, Da Capo Press, 2000.

[27] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Series in Artificial Intelligence, Prentice-Hall, 2nd edn., 2003.

[28] J. Searle, *Speech Acts*, Cambridge University Press, 1969.

[29] M. P. Singh, 'An ontology for commitments in multiagent systems', *Artif. Intell. Law*, **7**(1), 97–113, (1999).

[30] M. P. Singh, 'A social semantics for agent communication languages', in *Issues in Agent Comm.*, *LNCS* 1916, pp. 31–45. Springer, (2000).

[31] M. P. Singh, 'Distributed enactment of multiagent workflows: temporal logic for web service composition', in *Proc. of AAMAS'03*, pp. 907–914. ACM, (2003).

[32] M. P. Singh and A. K. Chopra, 'Correctness properties for multiagent systems', In *DALT* [1], pp. 192–207.

[33] P. Torroni, F. Chesani, P. Mello, and M. Montali, 'Social commitments in time: Satisfied or compensated', In *DALT* [1], pp. 228–243.

[34] M. B. van Riemsdijk, *Cognitive Agent Programming, a Semantic Approach*, Ph.D. dissertation, Utrecht University, the Netherlands, 2006.

[35] M. Winikoff, W. Liu, and J. Harland, 'Enhancing commitment machines', in *Proc. of DALT*, *LNCS* 3476, pp. 198–220, (2004).

[36] M. Wooldridge, *An Introduction to MultiAgent Systems*, John Wiley & Sons, 2002.

[37] P. Yolum and M. P. Singh, 'Commitment machines', in *Proc. of ATAL*, *LNCS* 2333, pp. 235–247. Springer, (2001).

[38] P. Yolum and M. P. Singh, 'Designing and executing protocols using the event calculus', in *Agents*, pp. 27–28, (2001).