Behavior-Oriented Commitment-based Protocols
Matteo Baldoni, Cristina Baroglio, and Elisa Marengo

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 openness 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


\( C(x, y, r, p) \) 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 \( \text{true} \). 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 \textit{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 \textit{decides} which actions to perform depending on the particular situation it is facing. Agents show a \textit{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 \textit{chair} of the assembly and the \textit{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 \textit{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 \textit{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 \textit{regulative} and the \textit{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 \textit{re-use} of actions in different contexts, a simpler \textit{customization} and \textit{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 \textit{openness}, \textit{interoperability}, and \textit{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 \textit{relate commitments} (more in general, literals) and \textit{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 \textit{constitutive} specification, defining the meaning of actions for all the agents in the system, and a \textit{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 \textit{constitutive specification} of the protocol, while the set of constraints \( C \), defined on \( F \) and on \( R \), forms the \textit{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 \textit{constitutive} specification of RONR is:

\[
\begin{align*}
(a) & \ \text{motion}(c,m) \iff \forall p\in P \ \text{CREATE}(C(c,p,cfv(m))) \\
(b) & \ \text{openDebate}(c,p,m) \iff \text{CREATE}(C(c,p,assignFloor(p,m))) \\
(c) & \ \text{refuseFloor}(p,c,m) \iff \text{refusedFloor}(p,m) \land \text{RELEASE}(C(c,p,assignFloor(p,m))) \\
(d) & \ \text{askFloor}(p,c,m) \iff \text{CREATE}(C(p,c,discussed(p,m))) \\
(e) & \ \text{recognition}(c,p,m) \iff \text{assignFloor}(p,m) \land \text{RELEASE}(C(c,p,assignFloor(p,m))) \\
(f) & \ \text{startTalk}(p,m) \iff \text{discussing}(p,m) \\
(g) & \ \text{stopTalk}(p,m) \iff \text{discussed}(p,m) \land \neg \text{discussing}(p,m) \land \neg \text{assignFloor}(p,m) \\
(h) & \ \text{timeOut}(c,p) \iff \text{discussed}(p,m) \land \neg \text{discussing}(p,m) \land \neg \text{assignFloor}(p,m) \land \text{RELEASE}(C(c,p,discussed(p,m))) \\
(i) & \ \text{cfv}(c,m) \iff \text{cfv}(m) \\
(j) & \ \text{vote}(p,m) \iff \text{voted}(p,m)
\end{align*}
\]

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
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 regulatory specifications. The regulative specification follows the grammar:

\[
\begin{align*}
C & \rightarrow (\text{Disj op Disj})^+ \\
\text{Disj} & \rightarrow \text{Conj} \lor \text{Disj} | \text{Conj} \\
\text{Conj} & \rightarrow \text{literal} \land \text{Conj} | \text{literal}
\end{align*}
\]

C is a set of constraints of the form \(A \text{ op } B\), where \(A\) and \(B\) are formulas of literals in disjunctive normal form (DNF) and \(\text{op}\) is one of the operators in Table 1; \(\text{literal}\) 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 (\(\circlearrowright\)), weak until (\(\trianglerighteq\)), 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 \(b\) becomes 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 \(a\) becomes true, \(b\) does not 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:

\[
\begin{align*}
c1: & \quad C(c,p,cfv(m)) \iff D(c,p,assignFloor(p,m)) \\
c2: & \quad C(p,m,assignFloor(p,m)) \iff D(p,m,assignFloor(p,m)) \\
c3: & \quad C(p,c,discuss(p,m)) \iff D(p,m,discussing(p,m)) \\
c4: & \quad assignFloor(p,m) \iff D(p,m,assignFloor(p,m)) \\
c5: & \quad D(p,m,assignFloor(p,m)) \iff assignFloor(p,m) \\
c6: & \quad D(p,m,discussing(p,m)) \iff assignFloor(p,m) \\
c7: & \quad D(p,m,discussing(p,m)) \iff assignFloor(p,m) \\
c8: & \quad c9: & \quad c8: & \quad c7: & \quad c6: & \quad c5: & \quad c4: & \quad c3: & \quad c2: & \quad c1: & \quad c0: & \quad c0: & \quad c10: & \quad c10: & \quad c9: & \quad c8: & \quad c7: & \quad c6: & \quad c5: & \quad c4: & \quad c3: & \quad c2: & \quad c1: & \quad c0: & \quad
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 not just a guideline because it is fundamental 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 $\leftrightarrow$ cfv(postpone(m)) and cfv(postpone(m)) $\rightarrow$ 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-) ex-

<table>
<thead>
<tr>
<th>Relation</th>
<th>Type</th>
<th>Positive LTL</th>
<th>Negative LTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation</td>
<td>base</td>
<td>A $\rightarrow$ B $\land$ (A $\supset$ B)</td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
</tr>
<tr>
<td>persistence</td>
<td></td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
</tr>
<tr>
<td>Co-existence</td>
<td>base</td>
<td>A $\rightarrow$ B $\land$ B $\rightarrow$ A</td>
<td>A $\models$ B $\land$ B $\models$ A</td>
</tr>
<tr>
<td>persistence</td>
<td></td>
<td>A $\models$ B $\land$ B $\models$ A</td>
<td>A $\models$ B $\land$ B $\models$ A</td>
</tr>
<tr>
<td>Response</td>
<td>base</td>
<td>A $\rightarrow$ B $\land$ (A $\models$ B)</td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
</tr>
<tr>
<td>persistence</td>
<td></td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
</tr>
<tr>
<td>Before</td>
<td>base</td>
<td>A $\rightarrow$ B $\land$ (A $\models$ B)</td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
</tr>
<tr>
<td>persistence</td>
<td></td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
</tr>
<tr>
<td>Cause</td>
<td>base</td>
<td>A $\rightarrow$ B $\land$ (A $\models$ B)</td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
</tr>
<tr>
<td>persistence</td>
<td></td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
</tr>
<tr>
<td>Premise</td>
<td>base</td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
</tr>
<tr>
<td>Immediately after</td>
<td>base</td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
<td>A $\models$ B $\land$ (A $\models$ B)</td>
</tr>
</tbody>
</table>

Table 1. 2CL constraint relations and their semantics in LTL.
ecutability of the actions. So, for instance in order to impose an or-
dering between the actions discuss a motion and obtain floor in the
RONR protocol, the action discuss a motion should have as a pre-
condition 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 con-
stitutive and the regulative specifications are indistinguishable in
the protocol, being both given by actions. The problem is that by do-
ing 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 specifi-
cation 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 univ-
ocally associated to it, as an effect of the action, and, then, to define
constraints (typically of kind premise, \(\prec\)) 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 prefer-
ence 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, giv-
ing 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 recog-
nize 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 con-
stitutive component but uses the declarative language ConDec for
representing business processes. Though not exactly corresponding
to interaction protocols, business processes specify the expected be-
havior of a set of interacting parties by constraining the execution of
their tasks. The regulative rules are a first-class element of this rep-
resentation. 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 ser-
vice 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: stand-
ing 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 al-
low 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 reg-
ulative 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 asso-
ciate 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 con-
straining 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 ac-
tions 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, how-
ever, 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 de-
finite, 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 specifi-
cation 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 pro-
ocols, indirectly ruling the execution of the actions. The regulative
specification is introduced because, in our opinion, the mere consti-
tutive specification of actions is not sufficient, because agents have
a behavior and this behavior makes them use actions in specific or-
ders. This order gives actions a supplementary meaning, that must
be taken into account in the interaction with the others. Since pro-
tocols are supposed to give the shared meaning of actions it is nec-
essary 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 particu-
lar, 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 in-
teroperability). In general, it is possible for agents to be compatibile
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 behav-
ior 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 behav-
or 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
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 same 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