

# Some Thoughts about Commitment Protocols (Position Paper)

Matteo Baldoni and Cristina Baroglio

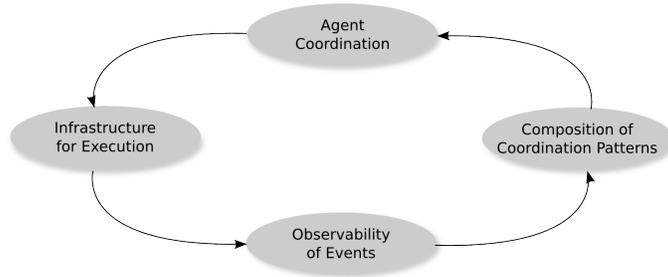
Università degli Studi di Torino  
Dipartimento di Informatica  
c.so Svizzera 185, I-10149 Torino (Italy)  
{matteo.baldoni,cristina.baroglio}@unito.it

**Abstract.** From the seminal paper by Singh [22], commitment protocols have been raising a lot of attention. The key feature of commitment protocols is their *declarative nature*, which allows specifying them in a way which abstracts away from any reference to the actual behavior of the agents. By doing so, commitment protocols respect the autonomy of agents. After more than ten years from the introduction of commitments, it is time to ask (i) if a “commitment to do something” is *the only kind of regulative norm*, that we need in order to give a social semantics to a physical action, and (ii) if they realize what they promised. In this position paper we discuss these points.

## 1 Introduction

Practical commitments lie at the level of *regulative* (or *preservative*) norms that, in turn, impact on the agents’ behavior, creating social expectations, that should not be frustrated. By a practical commitment, in fact, an actor (*debtor*) is committed towards another actor (*creditor*) to bring about something [9, 20], i.e. to act either directly or by persuading others so as to make a condition of interest become true. Due to their *social nature*, practical commitments are a powerful tool that helps to overcome the controversial assumptions of the mentalistic approach that mental states are verifiable and that agents are sincere. Moreover, they support an *observational semantics* for communication that allows verifying an agent’s compliance with its commitments based on observable behavior.

From the seminal paper by Singh [22], commitment protocols have been raising a lot of attention, see for instance [27, 17, 25, 12, 24, 14, 6]. The key feature of commitment protocols is their *declarative nature*, which allows specifying them in a way which abstracts away from any reference to the actual behavior of the agents, thus avoiding to impose useless execution constraints [28]. By doing so, commitment-based protocols respect the autonomy of agents because whatever action they decide to perform is fine as long as they accomplished their commitments, satisfying each others’ expectations. Now, after more than ten years from the introduction of commitments, it is time to ask (i) if a “commitment to do something” is *the only kind of regulative norm*, that we need in order to give a social semantics to a physical action, and (ii) if they realize what they



**Fig. 1.** The four considered intertwined aspects.

promised. To this aim, we think that there are four intertwined aspects to be considered (see Figure 1):

1. *Agent Coordination*: how to account for coordination patterns?
2. *Infrastructure for Execution*: which is the reference execution infrastructure?
3. *Observability of Events*: are events really observable by all agents?
4. *Composition of Coordination Patterns*: is composition influenced by the previous aspects?

## 2 Agent Coordination

Commitment protocols leave the choice of which action to execute and when, totally up to the single agents. From a more general perspective, they do not impose constraints on the possible evolutions of the social state. However, in many practical cases there is the need to capture regulative aspects of *agent coordination*. For instance, a customer and a merchant may *agree* that payment should be done before shipping but how to represent this socially agreed constraint in commitment protocols? When a similar coordination is desired by the parties, one feels the lack of the means for capturing them *as regulations* inside the protocol. Notice that the desired coordination patterns, though restricting the choices up to the agents, would not prevent flexibility because, for instance, it is not mandatory that payment and shipping are one next to the other. What matters is their relative order. More importantly, an agreed coordination pattern establishes the boundaries within which each party can exercise his/her own autonomy without compromising the aims for which the agreement was taken. Citing Dwight Eisenhower (State of the Union Address, Feb. 2, 1953) “*To be true to one’s own freedom is, in essence, to honor and respect the freedom of all others.*” As long as agents respect such constraints, they are free to customize the execution at their will, e.g. by interleaving the two actions with others (like sending a receipt or asking a quote for another item). This need is felt by the research community, see [4–6] for an overview.

When regulations are expressed, agents can individually check whether their behavior conforms to the specification [2]. But in order to guarantee to the others that one will act in a way that conforms to the regulation, an agent should *formally bind* its behavior to the regulation itself. The proposal in [3,6], for instance, allows the representation of temporal regulations imposed on the evolution of the social state, however, it does not supply a deontic semantics to the constraints. Therefore the agents' behavior is not formally bound to them. On the other hand, the REGULA framework [19] uses precedence logic to express temporal patterns that can be used as antecedent (or consequent) conditions inside commitments. Since patterns may involve various parties, the framework also introduces a notion of condition control and of commitment safety, in order to allow agents to reason about the advisability of taking a commitment. However, patterns are not generally expressed on the evolution of the social state but are limited to events.

### 3 Infrastructure for Execution and Observability of Events

Commitments were introduced to support *run-time verification* in contrast to the mentalistic approach but despite this, they still lack of a *reference infrastructure* that practically enables such a verification. Verification is supported by proposals like [1,10], although the authors do not draft an infrastructure, while commitment machines [28,25,23] have mainly been used to provide an operational semantics. Normative approaches, e.g. institutions [16,17], provide an answer but with some limitations. Indeed, they tend to implicitly assume a centralized vision, often realized by introducing a new actor, whose task is to monitor the interaction: the institution itself. This assumption is coherent with the fact that commitment protocols tend to assume that events are *uniformly observed* by all the agents although in the real world this seldom happens; for instance, communications tend to be point-to-point. For instance, consider an e-commerce seller, a supplier, and a client: the seller communicates with both the supplier and the client, who do not interact with one another. In other words, the interaction between each pair of actors is point-to-point and cannot be observed by the third party. We need the infrastructure to support *this* kind of interaction and to monitor, in *this* context, the on-going enactment, checking whether it respects all the regulative aspects – that the designer identified as relevant or that the agents agreed.

Chopra and Singh [11] addressed the issue of realizing an architecture that relaxes the centralization constraint by incorporating the notion of *commitment alignment*. In this way it becomes possible to answer questions like “how to decide whether agents are acting in a way that complies to the regulations or not?”, “How to know that an agent satisfied one of its commitments?” in contexts where events are not uniformly observable. Nevertheless, they relegated commitment alignment to the middleware, shielding the issue of *observability of events* from the agents and from the designer. Our claim is that this is a limitation and that

in many real-world situations it is more desirable to have the means of making clear who can access what information and who is accountable for reporting what event. This is especially true when the protocol allows the representation of coordination patterns: there is the need of mechanisms for expressing who can observe what, tracking which part of a pattern was already followed, which is left to be performed, who is in charge of the next moves, and so on. As a consequence, we think that the specification of the coordination patterns and the design of the infrastructure cannot leave out the *observability of events*, which plays a fundamental role *at the level of the protocol specification* and, for this reason, it should be captured by *first-class abstractions* and appropriate regulations. Such abstractions/regulations should be represented in a way that makes them *directly manipulable* by the agents [7].

## 4 Composition of Coordination Patterns

Most of the works concerning software engineering aspects of commitment protocol specification focus on the formal verification to help the protocol designer to get rid of or to enforce given behaviors, [26, 18, 8, 15, 14]. An aspect that is not to be underestimated is the realization of a development methodology for commitment protocols. The most relevant representative is the Amoeba methodology [13], which allows the design of commitment protocols and their composition into complex business processes. With respect to the aspects that we are discussing, this methodology, however, has two main limits. On the one hand, when two or more protocols are composed, the designer is requested to define a set of temporal constraints among events and of data flow constraints to combine the various parts. However, such constraints have neither a regulatory flavor nor a deontic characterization. On the other hand, since a wider number of roles are involved, which among actors of one protocol is entitled to (and physically can) observe events generated inside another protocol? The methodology does not explicitly account for this problem in the description of the various steps that compose it. For instance, suppose of composing a protocol that allows a merchant and a supplier to interact with one that allows the same merchant to interact with a customer. It is unrealistic to suppose that the client can observe events involving the supplier, even though after the composition both actors will play in the same protocol. Actually, it would be useful to incorporate in the protocol the means for letting the merchant tell the client that it received items from the supplier in a way that makes it accountable for its declarations.

## 5 Conclusive remarks

Commitments [21] are a powerful tool for creating communication and interaction standards with a solid and verifiable semantics, which is extremely important for dealing with open worlds, but to this aim there is the need of solving the issues that we have discussed.

1. *Agent Coordination: how to account for coordination patterns?* We claim that for accounting for coordination patterns there is the need of enriching the language for expressing commitment conditions (both antecedents and consequents) with temporal expressions in a way that shapes the desired interactions.
2. *Infrastructure for Execution: which is the reference execution infrastructure?* In our opinion, there is the need of reifying interaction protocols as first-class elements that can be manipulated and inspected by agents, rather than relegating them to the middleware.
3. *Observability of Events: are events really observable by all agents?* In the real world events are not uniformly observable by all the interacting parties. The seller-shipper-client example shows this fact in practice and proves that inside commitment protocols there is the need of specifying and managing objects like *claims, assertions, declarations, statements*, so typical in every day life as well as in programming languages. Indeed, by stating something about a state of things that the client cannot observe directly, the seller took a commitment, though not a practical one. A crucial limitation of the interaction protocols literature is that here the used commitments are always *practical*, meaning that they describe what the roles involved would bring about (e.g. a buyer commits to paying for some item). Practical commitments are limited to the debtor's own capabilities and powers (including persuading others), however, real scenarios often require the *account of some event*, without delegating the burden of making it happen to the agent who gives the account.
4. *Composition of Coordination Patterns: is composition influenced by the previous aspects?* All the above aspects should be supported by appropriate software engineering methodologies. This will have a positive impact on the acceptance of declarative approaches inside industrial settings.

## References

1. M. Alberti, M. Gavanelli, E. Lamma, F. Chesani, P. Mello, and P. Torroni. Compliance verification of agent interaction: a logic-based software tool. *Applied Artificial Intelligence*, 20(2-4):133–157, 2006.
2. M. Baldoni, C. Baroglio, A. K. Chopra, N. Desai, V. Patti, and M. P. Singh. Choice, Interoperability, and Conformance in Interaction Protocols and Service Choreographies. In *Proc. of AAMAS 2009*, pages 843–850, 2009. IFAAMAS.
3. M. Baldoni, C. Baroglio, and E. Marengo. Behavior-Oriented Commitment-based Protocols. In *Proc. of ECAI*, pages 137–142. IOS Press, 2010.
4. M. Baldoni, C. Baroglio, and E. Marengo. Constraints among Commitments: Regulative Specification of Interaction Protocols. In *Proc. of Int. Workshop on Agent Communication, AC 2010*, pages 2–18, Toronto, Canada, May 2010.
5. M. Baldoni, C. Baroglio, and E. Marengo. Commitment-based Protocols with Behavioral Rules and Correctness Properties of MAS. In *Post-Proc. of DALI 2010*, LNAI 6619, pages 60–77, Toronto, Canada, May 2011. Springer.
6. M. Baldoni, C. Baroglio, E. Marengo, and V. Patti. Constitutive and Regulative Specifications of Commitment Protocols: a Decoupled Approach. *ACM TIST, Spec. Iss. on Agent Communication*, 4(2), 2013.

7. M. Baldoni, C. Baroglio, E. Marengo, V. Patti, and A. Ricci. Back to the future: An interaction-oriented framework for social computing. In *First Int. Workshop on Req. Eng. for Social Computing, RESC*, pages 2–5. IEEE, 2011.
8. J. Bentahar, J.-J. Ch. Meyer, and W. Wan. Model checking communicative agent-based systems. *Knowl.-Based Syst.*, 22(3):142–159, 2009.
9. C. Castelfranchi. Commitments: From Individual Intentions to Groups and Organizations. In *Proc. of ICMAS*, pages 41–48, 1995.
10. F. Chesani, P. Mello, M. Montali, and P. Torroni. Commitment Tracking via the Reactive Event Calculus. In *Proc. of IJCAI*, pages 91–96. 2009.
11. A. K. Chopra and M. P. Singh. An Architecture for Multiagent Systems: An Approach Based on Commitments. In *Proc. of ProMAS*, LNAI 5919, pages 148–162, Heidelberg, 2009. Springer.
12. A. K. Chopra. *Commitment Alignment: Semantics, Patterns, and Decision Procedures for Distributed Computing*. PhD thesis, NCSU, Raleigh, NC, 2009.
13. N. Desai, A.K. Chopra, and M.P. Singh. Amoeba: A methodology for modeling and evolving cross-organizational business processes. *ACM TSEM*, 19(2), 2009.
14. M. El-Menshawy, J. Bentahar, and R. Dssouli. Verifiable Semantic Model for Agent Interactions Using Social Commitments. In *Proc. of LADS*, LNAI 6039, pages 128–152, 2010.
15. M. El-Menshawy, J. Bentahar, and R. Dssouli. Symbolic Model Checking Commitment Protocols Using Reduction. In *Post-proc. of DALT 2010*, LNAI 6619, pages 185–203. Springer, 2011.
16. N. Fornara. *Interaction and Communication among Autonomous Agents in Multiagent Systems*. PhD thesis, Univ. della Svizzera italiana, June 2003.
17. N. Fornara and M. Colombetti. Defining Interaction Protocols using a Commitment-based Agent Communication Language. In *Proc. of AAMAS*, pages 520–527. ACM, 2003.
18. A. Mallya and M. Singh. An algebra for commitment protocols. *Autonomous Agents and Multi-Agent Systems*, 14(2):143–163, 2007.
19. E. Marengo, M. Baldoni, C. Baroglio, A. K. Chopra, V. Patti, and M. P. Singh. Commitments with Regulations: Reasoning about Safety and Control in REGULA. In *Proc. of AAMAS*, pages 467–474, 2011.
20. M. P. Singh. An Ontology for Commitments in Multiagent Systems. *Artificial Intelligence and Law*, 7(1):97–113, 1999.
21. M. P. Singh. Community Standards for Agent Communication. Unpublished Draft, available at <http://www.csc.ncsu.edu/faculty/mpsingh/papers/drafts/Singh-AC-Manifesto.pdf>, July 2010.
22. M. P. Singh. Agent communication languages: Rethinking the principles. *IEEE Computer*, 31(12):40–47, 1998.
23. M. P. Singh. Formalizing Communication Protocols for Multiagent Systems. In *Proc. of IJCAI*, pages 1519–1524, 2007.
24. P. Torroni, F. Chesani, P. Mello, and M. Montali. Social Commitments in Time: Satisfied or Compensated. In *Proc. of DALT*, LNAI 5948, pages 228–243, 2010.
25. M. Winikoff, W. Liu, and J. Harland. Enhancing Commitment Machines. In *Post-proc. of DALT*, LNAI 3476, pages 198–220, 2005. Springer.
26. P. Yolum. Design time analysis of multiagent protocols. *Data Knowl. Eng.*, 63(1):137–154, 2007.
27. P. Yolum and M. P. Singh. Designing and Executing Protocols Using the Event Calculus. In *Agents*, pages 27–28, New York, NY, USA, 2001. ACM.
28. P. Yolum and M. P. Singh. Commitment Machines. In *Proc. of ATAL*, pages 235–247, 2002.