A Retrospective on the Reactive Event Calculus and Commitment Modeling Language

Paolo Torroni[1], Federico Chesani[1], Paola Mello[1], and Marco Montali[2]

[1] DISI, University of Bologna, Italy
`{paolo.torroni,federico.chesani,paola.mello}@unibo.it`
[2] KRDB, Free University of Bozen-Bolzano, Italy
`montali@inf.unibz.it`

Abstract. *Social commitments in time: Satisfied or compensated* was the title of a presentation given at the 7th DALT workshop edition [34] in which we proposed a layered architecture for modeling and reasoning about social commitments. We gave emphasis to modularity and to the need of accommodating certain temporal aspects in order for a commitment modeling framework to be flexible enough to adapt to diverse commitment theories, and expressive enough to model realistic scenarios. We grounded the framework on two formalisms: the Reactive Event Calculus ($\mathcal{REC}$) and the Commitment Modeling Language ($\mathcal{CML}$). In this retrospective, we review recent developments of this line of work, and discuss our contribution in a broader context of related research.

1 A Short Introduction to $\mathcal{REC}$ and $\mathcal{CML}$

Social commitments are a well-known concept in Multi-Agent Systems (MAS) research [8, 31]. They are commitments made from an agent to another agent to bring about a certain property. In broad terms, a social commitment represents the commitment that an agent, called *debtor*, has towards another agent, called *creditor*, to bring about some property or state of affairs, which is the *subject* of the commitment. In some instantiations of this idea, such as [18, 37], the subject of a commitment is a temporal logic formula.

Representing the commitments that the agents have to one another and specifying constraints on their interactions in terms of commitments provides a principled basis for agent interactions [35]. From a MAS modelling perspective, a role can be modelled by a set of commitments. For example, a seller in an online market may be understood as committing to its price quotes and a buyer may be understood as committing to paying for goods received. Commitments also serve as a natural tool to resolve design ambiguities. The formal semantics enables verification of conformance and reasoning about the MAS specifications [17] to define core interaction patterns and build on them by reuse, refinement, and composition.

Central to the whole approach is the idea of manipulation of commitments: their creation, discharge, delegation, assignment, cancellation, and release, since commitments are stateful objects that change in time as events occur. Time

C. Sakama et al. (Eds.): DALT 2011, LNAI 7169, pp. 120–127, 2012.

and events are, therefore, essential elements. Literature distinguishes between *base-level* commitments, written $\mathsf{C}(x, y, p)$, and *conditional* commitments, written $\mathsf{CC}(x, y, p, q)$ (x is the debtor, y is the creditor, and p/q are properties). $\mathsf{CC}(x, y, p, q)$ signifies that if p is brought out, x will be committed towards y to bring about q.

In our DALT 2009 paper *Social commitments in time: Satisfied or compensated* [34], we drew inspiration from work by Mallya *et al.* [24] and gave emphasis to temporal aspects of commitments. We wanted to propose an expressive enough notation, to be able to model commitment properties that have to be satisfied at specific time points or along specific intervals, and introduce a notion of compensation, with a mind on some scenarios in which social commitments may realistically be used. To this end, we identified a number of desiderata for social commitment frameworks. We then defined a new notation for commitments and commitment specification programs: the Commitment Modeling Language ($\mathcal{CML}$). Finally, we proposed an abstract commitment framework architecture and a concrete instance of it that supports $\mathcal{CML}$. In such an instance, temporal reasoning with commitments is operationalized using the Reactive Event Calculus ($\mathcal{REC}$), and various verification tasks can be accomplished thanks to an underlying declarative, computational logic-based framework.

The architecture proposed in [34] consists of four layers: a *user application* layer, a *commitment modeling* layer, a *temporal representation and reasoning* layer, and a *reasoning and verification* layer.

On the top layer, the user can define contracts or social interaction rules using commitments. Such definitions are based on a language provided by the layer below. The commitment modeling language is implemented using a temporal representation and reasoning framework, which is in turn built on top of a more general reasoning and verification framework, which lies at the bottom layer. It is important to rely on a formal framework that accommodates various forms of verification, because in this way commitments can be operationalized and the user can formally analyze commitment-based contracts, reason on the state of commitments, plan for actions needed to reach states of fulfillment, and track the evolution of commitments at run-time. Indeed, the underlying reasoning and verification layer must be powerful enough to accommodate temporal representation and reasoning.

Our proposal also included a concrete instance of such an architecture. We report it here (see Fig. 1). At the bottom of the stack lay a number of Prolog+CLP modules, which implement the SCIFF family of proof-procedures and provide the SCIFF language to the layer above [1]. The SCIFF framework is based on abductive logic programming and it consists of a declarative specification language and a family of proof-procedures for reasoning from SCIFF specifications. Some kinds of reasoning are: deduction, hypothetical reasoning, static verification of properties, compliance checking and run-time monitoring. In general, SCIFF comes in handy for a number of useful tasks in the context of agent interaction. Its main metaphor is that of *expectation* about events. A simple introduction to SCIFF and its usage is given in [35], where expectations are

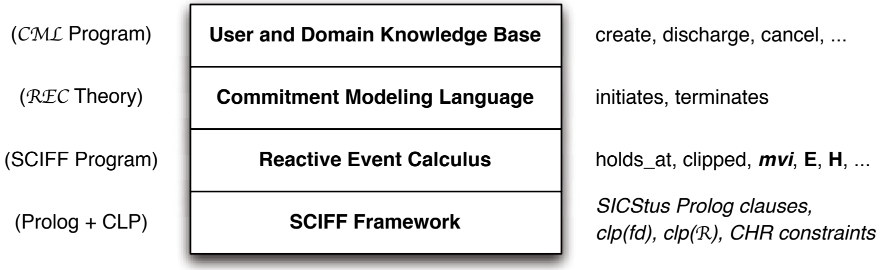

Fig. 1. Social commitment framework architecture

discussed in relation with commitments. The CLP solvers integrated in SCIFF can work with discrete and dense domains, depending on the application needs, and they are particularly useful for reasoning along the temporal dimension.

On top of the SCIFF layer we find the $\mathcal{REC}$: a SCIFF implementation of the $\mathcal{EC}$, which enables runtime verification [9]. In the third layer, the constructs that define $\mathcal{CML}$ are written by way of $\mathcal{REC}$ theories. Thus this layer provides the top layer with the language to write a $\mathcal{CML}$ program. The top layer consists of user and domain-dependent knowledge encoded into a $\mathcal{CML}$ program.

A sample $\mathcal{CML}$ program taken from [34] is the following one, which models a car rental contract inspired from a scenario due to [24]:

$$create(rent\_a\_car(T_c, T_e), \mathsf{C}(r, c, [T_c, T_c + 2days]great\_car)). \tag{1}$$

$$\begin{aligned} create(car\_broken(T_b), \mathsf{C}(r, c, [T_r]replace\_car)) \leftarrow \\ T_r \leq T_b + 24hours, holds([T_b]viol(\mathsf{C}(r, c, [T_s, T_e]great\_car), T_b)). \end{aligned} \tag{2}$$

Renting a car at time T_c until T_e creates a commitment that for 2 days as of T_c the car does not break down. The car breaking down at a time T_b creates a commitment that the car must be replaced within 24 hours of the incident, if the breakdown has caused a breach of commitment.

2 Recent Developments

Our implementation of $\mathcal{REC}$ on top of the SCIFF framework addressed an issue which had initially been introduced in [35], namely the reconciliation of commitments and expectations. Since the publication of [9], the $\mathcal{REC}$ framework has been fully implemented and is now distributed within the j-$\mathcal{REC}$ tool for run-time monitoring [10]. j-$\mathcal{REC}$, which embeds a tuProlog reasoner,[1] can be downloaded from `http://www.inf.unibz.it/~montali/tools.html#jREC`.

A significant and recent research direction, which is still subject of ongoing work, is monitoring and diagnosis of business contract exceptions. In [20–22],

---

[1] `http://sourceforge.net/projects/tuprolog`

Kafalı *et al.* study misalignment of commitments with temporal constraints. Misalignment is an undesirable situation in contract-regulated interactions, because it may bring about exceptions. To detect and therefore address occurrences of misalignment in an intrinsically distributed environment such as a multi-agent system or e-commerce setting, in [20] the authors present a diagnosis algorithm where agents reason based on the current states of their commitments. They also provide a method for automatic realignment, which can be applied by an agent when the diagnosis algorithm identifies a misalignment. $\mathcal{REC}$ is used to formalize the agent interactions in a delivery process scenario inspired from e-commerce.

As misalignments are typically due to mistakes in the delegation process, [21] and its extended version [22] focus on the notion of delegation. The authors propose a systematic classification of commitment delegation types, and identify *similarity* relations, to formalize connections among commitments. Understanding similarities enables handling exceptions in contract-regulated systems. In particular, it helps identifying possible reasons of exceptions by considering time-related commitments and "improper" ways of delegating such commitments, which may bring about inconsistent states. Again, the exception diagnosis framework is implemented in $\mathcal{REC}$.

The theoretical foundations of $\mathcal{REC}$, which we started to investigate in [9], were further explored in [11]. There we evaluate $\mathcal{REC}$ theoretically, discussing its formal properties and the use of negation, as well as from a practical perspective, by means of a examples dealing with quantitative temporal aspects, violations and compensations. On the application side, a recent survey [7] shows how $\mathcal{REC}$ has been applied to a variety of application domains, namely business process modeling, service-oriented computing, clinical guidelines and multi-agent systems. With respect to these different global computing domains, the survey identifies some challenges posed by concrete monitoring applications, showing how $\mathcal{REC}$ addresses them.

With respect to the multi-agent systems domain, we found that $\mathcal{REC}$ is successful not only in modeling and reasoning about e-commerce style contracts, but also in representing and reasoning upon the dynamic relations between agents and roles in multi-agent organizations [12] and in the context of agent-based simulation [13], for example to dynamically evaluate whether a running simulation is compliant with a given commitment-based contract, or to provide useful information to the interacting agents, helping them exhibit a compliant behaviour.

3 Related Work

We complete this retrospective with a brief survey on recent work by other authors, which is closely related to [9]. Two very relevant stuies by Yolum *et al.* were presented at DALT 2011 [19, 23]. The first one studies commitments in relation to each other. Following and extending our formalization of commitments based on $\mathcal{REC}/\mathcal{CML}$ [34, 9] Günay and Yolum identify key *conflict* relations among commitments. Conflict detection enables detecting a commitment violation before the actual violation occurs during agent interaction, and this knowledge can be used to guide an agent to avoid the violation. It can also be used during creation of multi-agent contracts to identify conflicts in the contracts. The authors implement their method in $\mathcal{REC}$. The second article, by Kafalı and Yolum, proposes a method to check if an agent's state complies with its projections, i.e., what they expect the outcome of a commitment-based contract to be, based on its content as well as their past experiences and the current world state. These projected states represent an agent's expectations from the future. The authors also propose a satisfiability relation, to check if an agent's state complies with its projections, and relate satisfiability with the occurrence of exceptions. The examples used in the paper show the importance of an explicit representation of (metric) time, especially in the subject of a commitment.

El Menshawy *et al.* [16, 26] addresses verification of social commitments and time following an approach alternative to our rule-based $\mathcal{REC}/\mathcal{CML}$ languages. The authors focus on the semantics of commitment operations, and propose a logical model based on an original extension of CTL^* with commitments and operations, and a new definition of assignment and delegation operations by considering the relationship between the original and new commitment contents. For the verification task, they rely on off-the-shelf symbolic model checkers such as NuSMV and MCMAS. The reader may be interested in comparing model checking-based and logic programming-based verification, especially in the context of domains that naturally lend themselves to declarative specifications, such as open multi-agent systems whose interactions are specified by social commitments. Montali *et al.* [28] present such a comparison, based on experimental results. Unfortunately, there is not much literature on this topic, also due to lack of benchmarks.

In a number of recent publications [4–6, 3, 25], Marengo *et al.* focus on the distinction between regulative and constitutive rules, and propose a new formalization of commitments where temporal regulations are incorporated as content of the commitments, using LTL as an underlying temporal language. This line of research suggests a possible future development of the $\mathcal{REC}/\mathcal{CML}$ framework, in which a explicit representation of time and the distinction between regulative and constitutive rules are combined in a unified framework. Some preliminary results on the formal relations between LTL and SCIFF are discussed in [27].

Frameworks for reasoning about events in time are rapidly gaining importance. In [2], Artikis *et al.* review representative approaches of logic-based event recognition, which is a key issue for many new applications that require efficient techniques for automated transformation of large data volumes into operational knowledge. A direction for future research is the evaluation of $\mathcal{REC}$'s reasoning efficiency, both theoretically and empirically. Although $\mathcal{REC}$ is implemented and used, such a systematic evaluation is still missing. Efficiency of temporal reasoning frameworks is an issue also for Patkos *et al.* [29, 30], who use the Jess rule-based system[2] to implement their Event Calculus reasoner. Urovi *et al.* [36] study selected versions of the Event Calculus to support efficient temporal reasoning without compromising the expressive power required to specify

---

[2] `http://www.jessrules.com`

norm-governed systems. Pros and cons of different implementations of the Event Calculus, taking into account reactivity, efficiency, intended application, and amenability to formal analysis, are also discussed by Chesani *et al.* in [11, 7].

Other interesting proposals for temporal representation and reasoning in the context of multi-agent systems are the action languages recently introduced by Pontelli *et al.*, applied to multi-agent planning [15] and commitment specification [32, 33]. In these last two articles, the authors show how the problem of verifying commitments or identifying pending commitments can be posed as queries to a narrative with commitments.

4 Conclusion

The abundance of recent proposals for modeling and representing events in time, especially in the context of commitments, demonstrates that this is a lively area, which has a good potential for further growth while posing at the same time interesting challenges. A reason for that is that there are many new applications requiring efficient and powerful techniques for describing and monitoring events in open domains. One such domain is multi-agent systems, where interactions can be described by way of commitment-based contracts. The $\mathcal{REC}/\mathcal{CML}$ framework we introduced in [34, 9] proposed an approach to these issues that aimed to be effective, both in terms of expressiveness (by accommodating metric time) and practical usability (by relying on procedures that make use of efficient, constraint-based solvers). Our initial work motivated further research, in contexts such as multi-agent contract exception handling, organization modeling and simulation.

In the future, we plan to integrate $\mathcal{REC}/\mathcal{CML}$ in a possible commitment-based middleware for agent development, such as that envisaged by Chopra and Singh [14]. There, instead of low-level communication primitives such as *send* and *receive*, the API would expose commitment-based operations such as *create*, *delegate*, *update*, and so on, and support listeners for commitment-related events. Another challenge we intend to take on is the systematic evaluation of our framework. However, performing an objective analysis of $\mathcal{REC}/\mathcal{CML}$ in relation with other commitment modeling and verification frameworks could be hard, due to a lack of suitable benchmarks. For this reason, in our works we took inspiration from what we considered realistic scenarios, and in [34] we attempted to define a number of desiderata for a commitment modeling framework.

References

1. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Verifiable agent interaction in abductive logic programming: the SCIFF framework. ACM Transactions on Computational Logic 9(4), 1–43 (2008)
2. Artikis, A., Skarlatidis, A., Portet, F., Paliouras, G.: Logic-based event recognition. Knowledge Engineering Review (to appear)
3. Baldoni, M., Baroglio, C., Marengo, E.: Behavior-oriented commitment-based protocols. In: Proc. 19th ECAI. Frontiers in Artificial Intelligence and Applications, vol. 215, pp. 137–142. IOS Press (2010)

4. Baldoni, M., Baroglio, C., Marengo, E.: Constraints among commitments: Regulative specification of interaction protocols. In: Proc. AC 2010, Toronto, Canada, pp. 2–18 (May 2010), `http://users.encs.concordia.ca/~bentahar/AC2010/AC2010.htm`
5. Baldoni, M., Baroglio, C., Marengo, E.: Commitment-Based Protocols with Behavioral Rules and Correctness Properties of MAS. In: Omicini, A., Sardina, S., Vasconcelos, W. (eds.) DALT 2010. LNCS, vol. 6619, pp. 60–77. Springer, Heidelberg (2011)
6. Baldoni, M., Baroglio, C., Marengo, E., Patti, V.: Constitutive and regulative specifications of commitment protocols: A decoupled approach. ACM Transactions on on Intelligent Systems and Technology (to appear)
7. Bragaglia, S., Chesani, F., Mello, P., Montali, M., Torroni, P.: Reactive event calculus for monitoring global computing applications. In: Essays in Honour of Marek Sergot: Computational Logic for Normative Systems. Springer, Heidelberg (to appear)
8. Castelfranchi, C.: Commitments: From individual intentions to groups and organizations. In: Proc. 1st ICMAS, pp. 41–48. The MIT Press (1995)
9. Chesani, F., Mello, P., Montali, M., Torroni, P.: Commitment tracking via the reactive event calculus. In: Proc. 21st IJCAI, pp. 91–96. AAAI (2009)
10. Chesani, F., Mello, P., Montali, M., Torroni, P.: A REC-based commitment tracking tool. In: 10th AI*IA/TABOO Italian Joint Workshop "From Objects to Agents", WOA 2009 (2009)
11. Chesani, F., Mello, P., Montali, M., Torroni, P.: A logic-based, reactive calculus of events. Fundamenta Informaticae 105(1-2), 135–161 (2010)
12. Chesani, F., Mello, P., Montali, M., Torroni, P.: Role Monitoring in Open Agent Societies. In: Jędrzejowicz, P., Nguyen, N.T., Howlet, R.J., Jain, L.C. (eds.) KES-AMSTA 2010, Part I. LNCS, vol. 6070, pp. 112–121. Springer, Heidelberg (2010)
13. Chesani, F., Mello, P., Montali, M., Torroni, P.: Monitoring time-aware commitments within agent-based simulation environments. Cybernetics and Systems 42(7), 546–566 (2011)
14. Chopra, A.K., Singh, M.P.: Elements of a Business-Level Architecture for Multiagent Systems. In: Braubach, L., Briot, J.-P., Thangarajah, J. (eds.) ProMAS 2009. LNCS, vol. 5919, pp. 15–30. Springer, Heidelberg (2010)
15. Dovier, A., Formisano, A., Pontelli, E.: An investigation of multi-agent planning in clp. Fundamenta Informaticae 105(1-2), 79–103 (2010)
16. El-Menshawy, M., Bentahar, J., Dssouli, R.: Verifiable Semantic Model for Agent Interactions Using Social Commitments. In: Dastani, M., El Fallah Segrouchni, A., Leite, J., Torroni, P. (eds.) LADS 2009. LNCS, vol. 6039, pp. 128–152. Springer, Heidelberg (2010)
17. Fisher, M., Bordini, R.H., Hirsch, B., Torroni, P.: Computational logics and agents: A road map of current technologies and future trends. Computational Intelligence 23(1), 61–91 (2007)
18. Fornara, N., Colombetti, M.: Operational specification of a commitment-based agent communication language. In: Proc. 1st AAMAS, pp. 536–542. ACM Press (2002)
19. Gunay, A., Yolum, P.: Detecting Conflicts in Commitments. In: Sakama, C., et al. (eds.) DALT 2011. LNCS (LNAI), vol. 7169, pp. 51–66. Springer, Heidelberg (2012)
20. Kafalı, Ö., Chesani, F., Torroni, P.: What Happened to My Commitment? Exception Diagnosis among Misalignment and Misbehavior. In: Dix, J., Leite, J., Governatori, G., Jamroga, W. (eds.) CLIMA XI. LNCS (LNAI), vol. 6245, pp. 82–98. Springer, Heidelberg (2010)
21. Kafali, O., Torroni, P.: Diagnosing commitments: delegation revisited (extended abstract). In: Proc. 10th AAMAS, IFAAMAS, pp. 1175–1176 (2011)

22. Kafalı, Ö., Torroni, P.: Social Commitment Delegation and Monitoring. In: Leite, J., Torroni, P., Ågotnes, T., Boella, G., van der Torre, L. (eds.) CLIMA XII 2011. LNCS, vol. 6814, pp. 171–189. Springer, Heidelberg (2011)
23. Kafali, O., Yolum, P.: A distributed treatment of exceptions in multiagent contracts (preliminary report). In: Proc. 9th DALT, pp. 65–78 (2011)
24. Mallya, A.U., Yolum, p., Singh, M.P.: Resolving Commitments among Autonomous Agents. In: Dignum, F.P.M. (ed.) ACL 2003. LNCS (LNAI), vol. 2922, pp. 166–182. Springer, Heidelberg (2004)
25. Marengo, E., Baldoni, M., Baroglio, C., Chopra, A.K., Patti, V., Singh, M.P.: Commitments with regulations: Reasoning about safety and control in REGULA. In: Proc. 10th AAMAS, IFAAMAS, pp. 843–850 (2011)
26. Menshawy, M.E., Bentahar, J., Qu, H., Dssouli, R.: On the verification of social commitments and time. In: Proc. 10th AAMAS, IFAAMAS, pp. 483–490 (2011)
27. Montali, M.: Specification and Verification of Declarative Open Interaction Models. A Logic-Based Approach. LNBIP, vol. 56, pp. 383–385. Springer, Heidelberg (2010)
28. Montali, M., Torroni, P., Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P.: Verification from Declarative Specifications Using Logic Programming. In: Garcia de la Banda, M., Pontelli, E. (eds.) ICLP 2008. LNCS, vol. 5366, pp. 440–454. Springer, Heidelberg (2008)
29. Patkos, T.: A formal theory for reasoning about action, knowledge and time. PhD thesis, Department of Computer Science, University of Crete, Greece (2010)
30. Patkos, T., Plexousakis, D.: Efficient epistemic reasoning in partially observable dynamic domains using hidden causal dependencies. In: Proc. 9th NRAC, pp. 55–62 (2011), `http://ijcai-11.iiia.csic.es/files/proceedings/W4-NRAC11-Proceedings.pdf#page=59`
31. Singh, M.P.: An ontology for commitments in multiagent systems: Toward a unification of normative concepts. Artificial Intelligence and Law 7, 97–113 (1999)
32. Son, T., Pontelli, E., Sakama, C.: Formalizing Commitments Using Action Languages. In: Sakama, C., et al. (eds.) DALT 2011. LNCS (LNAI), vol. 7169, pp. 67–83. Springer, Heidelberg (2012)
33. Son, T., Pontelli, E., Sakama, C.: Formalizing commitments using action languages. In: Proc. 10th Symposium on Logical Formalizations of Commonsense Reasoning. AAAI Spring Symposium Series. Stanford University (2011)
34. Torroni, P., Chesani, F., Mello, P., Montali, M.: Social Commitments in Time: Satisfied or Compensated. In: Baldoni, M., Bentahar, J., van Riemsdijk, M.B., Lloyd, J. (eds.) DALT 2009. LNCS, vol. 5948, pp. 228–243. Springer, Heidelberg (2010)
35. Torroni, P., Yolum, P., Singh, M.P., Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P.: Modelling interactions via commitments and expectations. In: Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models, pp. 263–284. IGI Global, Hershey (2009)
36. Urovi, V., Bromuri, S., Stathis, K., Artikis, A.: Run-time support for norm-governed systems. Technical Report CSD-TR-10-01, Royal Holloway, University of London, UK (2010), `http://golem.cs.rhul.ac.uk/TR/CSD-TR-10-01.pdf`
37. Yolum, P., Singh, M.: Flexible protocol specification and execution: applying event calculus planning using commitments. In: Proc. 1st AAMAS, pp. 527–534. ACM Press (2002)