An event-based model for contracts

Massimo Bartoletti Tiziana Cimoli G. Michele Pinna
Dipartimento di Matematica e Informatica, Università degli Studi di Cagliari, Italy
Roberto Zunino
DISI-Università degli Studi di Trento and COSBI, Italy

We introduce an event-based model for contracts. Our model extends event structures with a new relation, which faithfully captures the circular dependencies among contract clauses. We establish whether an agreement exists which respects all the contracts at hand (i.e. all the dependencies can be resolved), and we detect the obligations of each participant. The main technical contribution is a correspondence between our model and a fragment of the contract logic PCL [4]. More precisely, we show that the reachable events are exactly those which correspond to provable atoms in the logic. Despite of this strong correspondence, our model improves [4] by exhibiting a finer-grained notion of culpability, which takes into account the legitimate orderings of events.

1 Introduction

Contracts will play an increasingly important role in the specification and implementation of distributed systems. Since participants in distributed systems may be mutually distrusted, and may have conflicting individual goals, the possibility that a participant behaviour may diverge from the expected one is quite realistic. To protect themselves against possible misconducts, participants should postpone actual collaboration until reaching an agreement on the mutually offered behaviour. This requires a preliminary step, where each participant declares her promised behaviour, i.e. her *contract*.

A contract is a sort of assume/guarantee rule, which makes explicit the dependency between the actions performed by a participant, and those promised in return by the others. Event structures [15] can provide a basic semantic model for assume/guarantee rules, by interpreting the enabling $b \vdash a$ as the contract clause: "I will do a *after* you have done b". However, event structures do not capture a typical aspect of contracts, i.e. the capability of reaching an agreement when the assumptions and the guarantees of the parties mutually match. For instance, in the event structure with enablings $b \vdash a$ and $a \vdash b$, none of the events a and b is reachable, because of the circularity of the constraints. An agreement would still be possible if one of the parties is willing to accept a weaker contract. Of course, the contract "I will do b" (modelled as $\vdash b$) will lead to an agreement with the contract $b \vdash a$, but it offers no protection to the participant who offers it, e.g. it can be stipulated without having anything in return.

In this paper we introduce a model for contracts, by extending (conflict-free) event structures with a new relation $\Vdash$. The contract $a \Vdash b$ (intuitively, "I will do a if you *promise* to do b") reaches an agreement with the dual contract $b \Vdash a$, while protecting the participant who offers it. We formalise agreements as configurations where all the participants have reached their goals. We show that the problem of deciding if an agreement exists can be reduced to the problem of proving a suitable formula in (a fragment of) the contract logic PCL [4], where an effective decision procedure for provability exists.

Once an agreement has been found, the involved participants may safely cooperate by performing events. Indeed, we prove that — even in the presence of dishonest participants which do not respect their promises — either all the participants reach their goals, or some of them is *culpable* of not having

To appear in EPTCS.

performed her duties. A culpable participant may then be identified (and possibly punished). Also the problem of detecting duties and identifying culpable participants can be solved by exploiting provability of PCL. Notably, while PCL does not distinguish between the immediate duties and those that will only be required later on in a computation (all provable atoms are considered duties in PCL), the richer semantical structure of our model allows for a finer-grained notion of duties, which depend on the actual events already performed in a contract execution.

2 Contract model

A contract (Def. 1) comprises a set of events (ranged over by $a, b, \ldots$) and a set of participants (ranged over by $\mathsf{A}, \mathsf{B}, \ldots$). Each event is uniquely associated (by a labelling ℓ) to a participant. Events are constrained by two relations: $\vdash$ is the enabling relation of [15], while $\Vdash$ is a new relation, which allows to cope with circularity. For instance, $D \vdash e$ states that e may be performed *after* all the events in D have happened; instead, $D \Vdash e$ means that e may be performed either if D has already happened (similarly to $\vdash$), or possibly "on credit", on the promise that the events in D will be performed at some future time. The goals of each participant are indicated by the relation *ok*: $\mathsf{A}\ ok\ X$ means that A is satisfied if *all* the events in X have happened. The composition of contracts is defined component-wise, provided that events are uniquely associated to participants.

Definition 1. *A contract γ is a 6-tuple $(\mathcal{E}, \mathcal{A}, \ell, ok, \vdash, \Vdash)$, where:* (1) *$\mathcal{E}$ is a finite set of* events*;* (2) *$\mathcal{A}$ is a finite set of* participants*;* (3) *$\ell : \mathcal{E} \to \mathcal{A}$ associates each event to a participant;* (4) *$ok \subseteq \mathcal{A} \times \mathcal{P}(\mathcal{E})$ is the* fulfillment *relation, such that $\mathsf{A}\ ok\ X \wedge X \subseteq Y \implies \mathsf{A}\ ok\ Y$;* (5) *$\vdash \subseteq \mathcal{P}(\mathcal{E}) \times \mathcal{E}$ is the* enabling relation*, such that $X \vdash e \wedge X \subseteq Y \implies Y \vdash e$;* (6) *$\Vdash \subseteq \mathcal{P}(\mathcal{E}) \times \mathcal{E}$ is the* circular enabling *relation.*

Example 2. *Suppose there are three kids who want to play together. Alice has a toy airplane, Bob has a bike, while Carl has a toy car. Each of the kids is willing to share his toy, but they have different constraints: Alice will lend her airplane only* after *Bob has allowed her ride his bike; Bob will lend his bike only after he has played with Carl's car; Carl will lend his toy car if the other two kids promise that they will eventually let him play with their toys. These constraints are modelled by the following contract γ, where in the relation ok we only indicate the minimal goals, and the components $\mathcal{E}$, $\mathcal{A}$, and ℓ are omitted for brevity:*

$$\{b\} \vdash a \qquad \{c\} \vdash b \qquad \{a,b\} \Vdash c \qquad\qquad \mathsf{A}\ ok\ \{b\} \qquad \mathsf{B}\ ok\ \{c\} \qquad \mathsf{C}\ ok\ \{a,b\}$$

In the previous example, it is crucial that Carl's contract allows the event c to happen "on credit" before the other events are performed. We shall show that this leads to an agreement among the participants, while no agreement exists were Carl requiring $\{a,b\} \vdash c$ (cf. Ex. 5).

In Def. 3 we refine the notion of configuration of [15], so to deal with the new $\Vdash$-enablings. A set of events C is a configuration if its events can be ordered in such a way that each event $e \in C$ is either $\vdash$-enabled by its predecessors, or it is $\Vdash$-enabled by some subset D of C. Configurations play a crucial role, as they represent sets of events where all the debts have been honoured.

Definition 3. *For all contracts γ, we say that $C \subseteq \mathcal{E}$ is a* configuration of γ *iff*

$$\forall e \in C.\ \exists e_0, \ldots e_n \in C.\ \big(e_n = e \wedge \forall i \leq n.\ (\{e_0, \ldots, e_{i-1}\} \vdash e_i \vee \exists D \subseteq C.\, D \Vdash e_i)\big)$$

The set of all configurations of γ is denoted by $\mathcal{F}(\gamma)$.

Example 4. *Not all sets of events are also configurations. For instance, in the contract with $a \Vdash b$ and $b \Vdash a$, the sets $\emptyset$ and $\{a,b\}$ are configurations (in the latter, the use of $\Vdash$ allows for resolving the circular dependency between a and b), while $\{a\}$ and $\{b\}$ are not.*

Example 5. *The contract γ of Ex. 2 has configurations $\emptyset$ and $\mathcal{E} = \{a,b,c\}$. Note that if Carl replaces his contract with $\{a,b\} \vdash c$, then $\mathcal{E}$ no longer belongs to $\mathcal{F}(\gamma)$.*

Following the examples above we observe that, differently from other event-based models, if C is a configuration, not necessarily $X \subseteq C$ is a configuration as well. Hereafter, subsets of $\mathcal{E}$ are called *states*, regardless they are configurations or not.

The monotonicity of the relation *ok* models the fact that once a contract has been fulfilled (i.e. a configuration is reached where all participants say *ok*), additional events can be neglected. To cope with that, our contracts have no conflicts (unlike [15]). Consequently, the union of two configurations is always a configuration.

Lemma 6. *For all contracts γ, if $C \in \mathcal{F}(\gamma)$ and $D \in \mathcal{F}(\gamma)$, then $C \cup D \in \mathcal{F}(\gamma)$.*

Notice that an event e may be added to a configuration C only if either $C \vdash e$, or $D \Vdash e$ for some $D \subseteq C$. Otherwise, $C \cup \{e\}$ is not necessarily a configuration. Compositional reasoning on sets of events (not necessarily configurations) can be done as sketched in the proof of Th. 15, by keeping track of the events taken "on credit".

An event is *reachable* when it belongs to a configuration; a state X is reachable if every event in X is reachable. On the converse, whenever a set X is reachable, there always exists a configuration that contains X, as by Lemma 6 configurations are closed by union. Also, the set comprising all the reachable event is a configuration (actually, it is the maximal one).

Lemma 7. *Let $X \subseteq \mathcal{E}$ be a reachable state. Then, $\exists C \in \mathcal{F}(\gamma).\ X \subseteq C$.*

Lemma 8. *For all γ, $C = \bigcup\{e \in \mathcal{E} \mid e \text{ is reachable}\} \in \mathcal{F}(\gamma)$, and $\nexists C' \in \mathcal{F}(\gamma)$ such that $C' \supsetneq C$.*

2.1 Agreements

An *agreement* (Def. 9) is a configuration of a contract where all the participants have reached their individual goals. E.g., the configuration $\mathcal{E} = \{a,b,c\}$ is an agreement on the contract γ of Ex. 2, since P *ok* $\mathcal{E}$ holds for $P \in \{\mathsf{A},\mathsf{B},\mathsf{C}\}$ by monotonicity of *ok*.

Definition 9. *An* agreement *on γ is a configuration $C \in \mathcal{F}(\gamma)$ such that $\forall \mathsf{A} \in \mathcal{A} : \mathsf{A}$ *ok* C.*

We now establish the duties of a participant in a state where some events X have been performed. Although several different definitions of duties are possible, the common factor of any reasonable definition is that, in the absence of duties, all the participants must have reached their goals (see Th. 13). Here we focus on a definition of duties where $\vdash$ is prioritized over $\Vdash$, i.e. an event may be performed on credit only if no other ways are possible. More precisely, an event e belongs to $\mathit{duties}(\mathsf{A},X)$ if (i) e is not already present in X, but is in some configuration C, (ii) $\ell(e) = \mathsf{A}$, and (iii) either e is $\vdash$-enabled by X, or, if no $\vdash$-enablings are possible from X, then e is $\Vdash$-enabled by some events in $C \cup X$.

Definition 10. *For all A, for all X, we define $\mathit{duties}(\mathsf{A},X)$ as the set of events $e \notin X$ such that $\ell(e) = \mathsf{A}$ and there exists $C \in \mathcal{F}(\gamma)$ such that $e \in C$, and either $X \vdash e$ or $\nexists e' \in C \setminus X : X \vdash e' \wedge \exists D \subseteq C \cup X : D \Vdash e$. A participant A is* culpable *in X when A has some duties in X.*

Example 11. *Recall the contract γ of Ex. 2. By Def. 10, in state $\emptyset$ only participant C is culpable, with $\mathit{duties}(\mathsf{C},\emptyset) = \{c\}$; in $\{c\}$ only B is culpable, with $\mathit{duties}(\mathsf{B},\{c\}) = \{b\}$; finally, in $\{b,c\}$ only A is culpable, with $\mathit{duties}(\mathsf{A},\{b,c\}) = \{a\}$.*

Example 12. *Let γ be a contract with $\{a_0,a_1\} \Vdash a_2$, $\{a_0,a_2\} \Vdash a_1$, $\{a_1,a_2\} \vdash a_3$, and $\emptyset \vdash a_0$, where $\ell(a_i) = \mathsf{A}_i$ for $i \in [0,3]$. We have that only A_0 is culpable in $\emptyset$; only A_1 and A_2 are culpable in $\{a_0\}$; only A_1 is culpable in $\{a_0,a_2\}$; only A_2 is culpable in $\{a_0,a_1\}$; only A_3 is culpable in $\{a_0,a_1,a_2\}$; finally, no one is culpable in $C = \{a_0,a_1,a_2,a_3\} \in \mathcal{F}(\gamma)$.*

The following theorem establishes that it is safe to execute contracts after they have been agreed upon. More precisely, in each state X of the contract execution, either all the participant goals have been fulfilled, or some participant is culpable in X. Note that, in consequence of Def. 10, a participant can always exculpate herself by performing some of her duties. This is because, if $D = \mathit{duties}(\mathsf{A},X)$ is not empty, participant A is always allowed to perform all the events in D, eventually reaching a state where she is not culpable (note also that in the maximal state $\mathcal{E}$ no one is culpable).

Theorem 13. *If an agreement on γ exists, then for all participants $\mathsf{A} \in \mathcal{A}$, and for all $X \subseteq \mathcal{E}$, either A ok$X$, or some participant is culpable in X.*

2.2 On deciding agreements

The problem of deciding if an agreement exists on some contract γ is reduced below to the problem of proving formulae in the contract logic PCL [4]. A comprehensive presentation of PCL is beyond the scope of this paper, so we give here a brief overview, and we refer the reader to [4, 3] for more details.

PCL extends intuitionistic propositional logic IPC with the connective $\twoheadrightarrow$, called *contractual implication*. Differently from IPC, a contract $\mathsf{b} \twoheadrightarrow \mathsf{a}$ implies a not only when b is true, like IPC implication, but also in the case that a "compatible" contract, e.g. $\mathsf{a} \twoheadrightarrow \mathsf{b}$, holds. Also, PCL is equipped with an indexed lax modality *says* , similarly to the one in [11].

The Hilbert-style axiomatisation of PCL extend that of IPC with the following axioms:

$$\top \twoheadrightarrow \top \qquad \phi \to (A \mathit{\ says\ } \phi)$$
$$(\phi \twoheadrightarrow \phi) \to \phi \qquad (A \mathit{\ says\ } A \mathit{\ says\ } \phi) \to A \mathit{\ says\ } \phi$$
$$(\phi' \to \phi) \to (\phi \twoheadrightarrow \psi) \to (\psi \to \psi') \to (\phi' \twoheadrightarrow \psi') \qquad (\phi \to \psi) \to (A \mathit{\ says\ } \phi) \to (A \mathit{\ says\ } \psi)$$

In [4] a Gentzen-style proof system is given which enjoys cut elimination and the subformula property. The decidability of the entailment relation $\vdash_{\mathrm{PCL}}$ is a direct consequence of these facts.

In Def. 14 we show a translation from contracts into PCL formulae. In particular, our mapping is a bijection into the fragment of PCL (called 1N-PCL) which comprises atoms, conjunctions, says, and non-nested (standard/contractual) implications.

Definition 14. *The mapping $[\cdot]$ from contracts into* 1N-PCL *formulae is defined as follows:*

$$[(D_i \circ a_i)_i] = \textstyle\bigwedge_i [D_i \circ a_i] \qquad \mathit{where}\ [\circ] = \begin{cases} \to & \mathit{if} \circ = \vdash \\ \twoheadrightarrow & \mathit{if} \circ = \Vdash \end{cases}$$
$$[\{d_i \mid i \in \mathcal{I}\} \circ a] = \ell(a) \mathit{\ says\ } (\textstyle\bigwedge_{i\in\mathcal{I}} \ell(d_i) \mathit{\ says\ } d_i)[\circ]\ a$$

Theorem 15. *For all contracts γ, for all events e, e is reachable in γ iff $[\gamma] \vdash_{\mathrm{PCL}} \ell(e) \mathit{\ says\ } e$.*

Proof. (*Sketch*) We extend the definition of configuration, by allowing events performed "on credit", in the absence of their premises. We say that $C \subseteq \mathcal{E}$ is an *X-configuration* of γ iff $X \subseteq C$ and

$$\forall e \in C.\ \exists e_0, \ldots e_n \in C.\ e_n = e \wedge \forall i \leq n.\ \big(e_i \in X \vee \{e_0,\ldots,e_{i-1}\} \vdash e_i \vee \exists D \subseteq C.\ D \Vdash e_i\big)$$

Notice that Def. 3 is the special case when $X = \emptyset$. An event e is X-reachable if it belongs to some X-configuration. For all X, we define the set $\mathcal{R}(X)$ by the following inference rules:

$$\frac{D \vdash e \quad D \subseteq \mathcal{R}(X)}{e \in \mathcal{R}(X)} \qquad \frac{D \Vdash e \quad D \subseteq \mathcal{R}(X \cup \{e\})}{e \in \mathcal{R}(X)} \qquad \frac{e \in X}{e \in \mathcal{R}(X)}$$

The set $\mathcal{R}(X)$ is used as a bridge in proving that e is X-reachable iff $X, [\gamma] \vdash_{\text{PCL}} e$. We prove first that $\mathcal{R}(X)$ contains exactly the X-reachable events, and then (by induction on the depth of the derivations) that $X, [\gamma] \vdash_{\text{PCL}} e$ iff $e \in \mathcal{R}(X)$. To do that, it is helpful to exploit the following property of $\mathcal{R}$: for all X, Y, Z, if $Z \subseteq \mathcal{R}(Y)$ then $\mathcal{R}(X \cup Z) \subseteq \mathcal{R}(X \cup Y)$. □

The following theorem exploits the decidable proof system of PCL (actually, the much simpler one of $1N$-PCL is enough) to devise a decision procedure for agreements. Concretely, one can use the decision procedure of $1N$-PCL to compute the set C of reachable events. Then, an agreement exists iff each principal A has some goals contained in C.

Theorem 16. *A contract γ admits an agreement iff:*

$$\forall \mathsf{A} \in \mathcal{A}.\ \exists G \subseteq \mathcal{E}.\ \big(\mathsf{A}\ ok\ G \wedge \forall e \in G : [\gamma] \vdash_{\text{PCL}} \ell(e)\ says\ e\big)$$

Proof. ($\Rightarrow$) Let C be an agreement on γ, and let $\mathcal{A} = \{\mathsf{A}_i\}_i$. For all i, $\mathsf{A}_i\ ok\ C$ for all i, by Def. 9. By definition of *ok*, there exist $G_i \subseteq C$ such that $\mathsf{A}_i\ ok\ G_i$. Since $G_i \subseteq C \in \mathcal{F}(\gamma)$, then G_i is reachable. Therefore, by Theorem 15, $[\gamma] \vdash_{\text{PCL}} \ell(e)\ says\ e$, for all $e \in G_i$.

($\Leftarrow$) Let $\mathcal{A} = \{\mathsf{A}_i\}_i$, and let $\{G_i\}_i$ be such that $\mathsf{A}_i\ ok\ G_i$ and $[\gamma] \vdash_{\text{PCL}} \ell(e)\ says\ e$ for all i and for all $e \in G_i$. By Theorem 15, each G_i is reachable. By Lemma 7, for all i there exists $C_i \in \mathcal{F}(\gamma)$ such that $C_i \supseteq G_i$. By Lemma 6, $C = \bigcup_i C_i$ is an agreement on γ. □

Finally, note that also $duties(\mathsf{A}, X)$ can be computed by exploiting the correspondence with PCL. More precisely, we use $\vdash_{\text{PCL}}$ to compute the set of all reachable events, so obtaining the maximal configuration (Lemma 8), and then to compute $D \vdash e$ as prescribed by Def. 10.

3 Related work and Conclusions

We have proposed an event-based model for contracts, featuring an effective procedure for deciding when agreements exist, and which are the duties of the participants. We depart from the common principle that contracts are always respected after they are stipulated: in our model, promises might not be always maintained, and when they are not, culpable participants can be identified.

A concrete usage scenario of our contract model is a protocol for exchanging, agreeing upon, and executing contracts. In the initial phase of the protocol, a special participant T acts as a contract broker, which collects the contracts from all the participants. Then, T looks for possible agreements on subsets of the contracts at hand. After an agreement on γ has been found, T shares a session with the participants in γ. As long as the goals of some participant have not been fulfilled, T notifies the duties to each culpable participant. Variants of this protocol are possible which dispose T from some of his tasks. Notice that reaching an agreement is an essential requirement for the security of this protocol: if an untrusted contract broker claims to have found an agreement when there is none, then Theorem 13 no longer applies, and a situation is possible where a participant has not reached her goals, but no one is culpable. Participants can still protect themselved against untrusted brokers, by always requiring in their contracts the suitable preconditions. This protocol can be formally described in the process calculus CO_2 [2]. This would

require to specialise the abstract contract model of CO_2 to the contracts presented in this paper, and, accordingly, to make the observables in fuse/ask prefixes correspond to agreements/duties, respectively.

Contracts have been investigated using a variety of models, e.g. c-semirings [6, 7, 10], behavioural types [5, 8, 9], logics [1, 14], *etc*. All these models do not explicitly deal with the circularity issue, which instead is the focus of this paper. Circularity is dealt with at a logical (proof-theoretic) level in [4]; the relation between reachability in our model and provability in the logic of [4] is stated by Theorem 15. Compared to [4], our model features a finer notion of duties: while [4] focusses on reachable events, Def. 10 singles out which events must be performed in a given state, by interpreting $D \vdash e$ as "I will do *e after D* has been done". In [13] a trace-based model for contracts is defined. Similarly to ours, a way is devised for blaming misconducts, also taking into account time contraints. However, [13] is not concerned in how to reach agreements, so the modeling of mutual obligations (circularity) is neglected. It seems interesting to extend our model with temporal deadlines, which would allow for a tighter notion of agreement, and, more in general, with soft constraints, which could be used to model QoS requirements.

In [12] a generalization of prime event structures is proposed where a *response* relation (denoted with $\bullet\!\!\rightarrow$) is used to characterize the accepting traces as those where, for each $a \bullet\!\!\rightarrow b$, if a is present in the trace, then b eventually occurs after a. The response relation bears some resemblance with our $\Vdash$ relation, but there are some notable differences. First, having $a \Vdash b$ does not necessarily imply that a configuration containing a must contain also b (another enabling could have been used), whereas $a \bullet\!\!\rightarrow b$ stipulates that once one has a in an accepting configuration, then also b must be present. Indeed, an enabling $a \Vdash b$ can be neglected, whereas $a \bullet\!\!\rightarrow b$ must be used. Also, augmenting the number of $\Vdash$-enablings increases the number of configurations, while adding more response relations reduces the number of accepting configurations of the event structure. Finally, [12] deals with conflicts, while we have left this issue for future investigation.

Acknowledgments. This work has been partially supported by by Aut. Region of Sardinia under grants L.R.7/2007 CRP2-120 (Project TESLA) and CRP-17285 (Project TRICS).

References

[1] A. Artikis, M. J. Sergot, and J. V. Pitt. Specifying norm-governed computational societies. *ACM Trans. Comput. Log.*, 10(1), 2009.

[2] M. Bartoletti, E. Tuosto, and R. Zunino. Contracts in distributed systems. In *ICE*, 2011.

[3] M. Bartoletti and R. Zunino. A logic for contracts. Technical Report DISI-09-034, DISI - Univ. Trento, 2009.

[4] M. Bartoletti and R. Zunino. A calculus of contracting processes. In *LICS*, 2010.

[5] M. Bravetti and G. Zavattaro. Towards a unifying theory for choreography conformance and contract compliance. In *Software Composition*, 2007.

[6] M. G. Buscemi and H. C. Melgratti. Transactional service level agreement. In *TGC*, 2007.

[7] M. G. Buscemi and U. Montanari. CC-Pi: A constraint-based language for specifying service level agreements. In *ESOP*, 2007.

[8] S. Carpineti and C. Laneve. A basic contract language for web services. In *ESOP*, 2006.

[9] G. Castagna, N. Gesbert, and L. Padovani. A theory of contracts for web services. *ACM Transactions on Programming Languages and Systems*, 31(5), 2009.

[10] G. L. Ferrari and A. Lluch-Lafuente. A logic for graphs with QoS. *ENTCS*, 142, 2006.

[11] D. Garg and M. Abadi. A modal deconstruction of access control logics. In *FoSSaCS*, 2008.

[12] T. T. Hildebrandt and R. R. Mukkamala. Declarative event-based workflow as distributed dynamic condition response graphs. In *PLACES*, volume 69 of *EPTCS*, 2010.

[13] T. Hvitved, F. Klaedtke, and E. Zălinescu. A trace-based model for multiparty contracts. *JLAP*. To appear.

[14] C. Prisacariu and G. Schneider. A formal language for electronic contracts. In *FMOODS*, 2007.

[15] G. Winskel. Event structures. In *Advances in Petri Nets*, pages 325–392, 1986.