# Checking Correctness of Business Contracts via Commitments[*]

Nirmit Desai
NC State University
Raleigh, NC 27695-8206
nvdesai@ncsu.edu

Nanjangud C. Narendra
IBM India Research Lab
Bangalore, India
narendra@in.ibm.com

Munindar P. Singh
NC State University
Raleigh, NC 27695-8206
singh@ncsu.edu

## ABSTRACT

Business contracts tend to be complex. In current practice, contracts are often designed by hand and adopted by their participants after, at best, a manual analysis. This paper motivates and formalizes two aspects of contract correctness from the perspective of the preferences of the agents participating in them. A contract is *safe* for a participant if participating in the contract would not leave the participant worse off than otherwise. More strongly, a contract is *beneficial* to a participant if participating in the contract would leave the participant better off than otherwise.

This paper seeks to partially automate reasoning about the correctness of formally modeled business contracts. It represents contracts formally as a set of commitments. It motivates constraints on how cooperative agents might value the various states of commitments. Further, it shows that such constraints are consistent and promote cooperation. Lastly, it presents algorithms for checking the safety and guaranteed benefits of a contract.

## Categories and Subject Descriptors

K.4.4 [**Electronic commerce**]: Distributed commercial transactions; I.2.11 [**Distributed artificial intelligence**]: Multiagent systems

## General Terms

Design, Economics

## Keywords

Contract verification, agreement modeling

## 1. INTRODUCTION

Interorganizational business interactions are typically defined by *(business) contracts*. A contract describes the roles and responsibilities of its participants, along with the typical value exchanges that take place during contract enactment. In current practice, contracts are defined in natural language, and are often ambiguous. Given the size and complexity of business contracts, manual verification is both expensive and error prone. Incorrect contracts, not being compatible with the participants' preferences, are either subverted or carried out at some loss. Further, the risk of hidden hazards in contracts adds friction to the economy, thus preventing potential gains in trade.

As a motivating example, let's consider a real-life business contract [6]. Briefly, this contract is an agreement among Foamex, AMFS, Foamtec, and a Customer. Foam products are to be manufactured in Singapore and shipped to the Customer by AMFS. AMFS proposes to obtain raw materials from Foamex, ship them to Foamtec, obtain the finished product from Foamtec, and ship the product to the Customer. The contract merely states the terms and conditions under which the interorganizational interactions occur. It may turn out to be unsafe or not beneficial for a participant.

A participant cannot easily determine (a) whether it would be beneficial or safe to enter into this contract and (b) what additional constraints it might place on its interactions to ensure safety and obtaining a benefit. Whereas contracts often list failure conditions and any associated penalties, a participant would like to ensure that the contract is correct from its point of view, i.e., it is adequately protected. Accordingly, this paper addresses the important question of *contract correctness*: given (partial) knowledge of an agent's preferences, would it be safe or beneficial for that agent to enter into a specific contract?

We represent a contract as a collection of the participants' *commitments* toward each other [15, 17]. Thus we understand the interactions that occur during the enactment of a contract in terms of how they affect the participants' commitments according to a specific life cycle (described shortly). Additional constraints on interactions are captured via a *protocol*, understood as a set of coordination requirements.

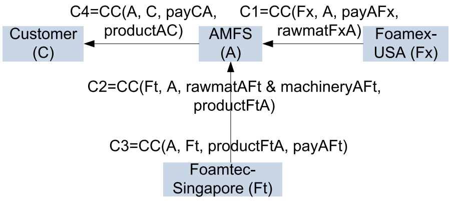
C4=CC(A, C, payCA, productAC)
C1=CC(Fx, A, payAFx, rawmatFxA)
Customer (C)
AMFS (A)
Foamex-USA (Fx)
C2=CC(Ft, A, rawmatAFt & machineryAFt, productFtA)
C3=CC(A, Ft, productFtA, payAFt)
Foamtec-Singapore (Ft)

**Figure 1: Commitments in the Foamtec contract**

Figure 1 depicts the Foamtec contract via commitments. Here, $\mathsf{CC}(x, y, p, q)$ means that $x$ is committed $y$ to bringing about $q$ if $p$ is brought about. A contract would thus be enacted via state changes on commitments. Each such change is valued (potentially differently) by the participants. For example, AMFS' paying Foamex would garner a positive value for Foamex but a negative value for AMFS.

The contributions of this paper are to the engineering and analysis of contracts. It studies the correctness of contracts from the perspective of an individual participant. It proposes algorithms for de-

[*] With partial support from the US National Science Foundation under grant IIS-0139037.

**Cite as:** Checking Correctness of Business Contracts via Commitments, Nirmit Desai, Nanjangud C. Narendra, and Munindar P. Singh, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. XXX-XXX.

termining the valuation criteria for a participant under which a contract is, respectively, safe or beneficial for that participant. These algorithms are implemented in a prototype design tool, using which a contract designer or agent implementer can explore the space of contracts and the protocols that enact them. Although this paper's subject matter touches upon theories of games and rationality, it makes no general contribution to those areas. Instead, it rationality uses as a motivation and provides a basis for a soundness test. In particular, we show that if our algorithms produce a solution, then at least one pure-strategy Nash equilibrium exists.

## Organization

Section 2 summarizes the key background on the commitment life cycle and valuation constraints. Section 3 introduces new valuation constraints, shows a model for the constraints, and formalizes definitions of correctness criteria. Section 4 presents our algorithms for checking correctness. Section 5 discusses the related work and some directions for future work.

# 2. BACKGROUND

This section reviews key background relating to commitments and rationality, which informs our technical approach.

## 2.1 Commitments

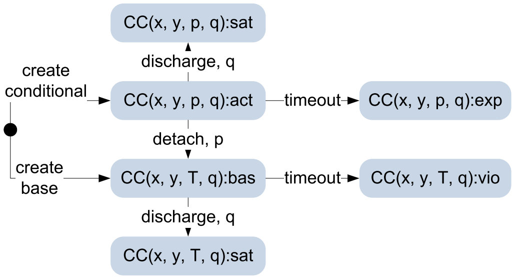

**Figure 2: Life cycle of commitments**

$\mathsf{CC}(x, y, p, q)$ denotes that $x$ is committed (roughly, obligated) to $y$ to bringing about $q$ if $p$ holds. Here $p$ is called the precondition and $q$ the condition of the commitment. If the precondition is $\mathsf{T}$ (true), it is a base commitment where the debtor is absolutely committed to bringing about the condition. Otherwise, it is a conditional commitment. For understanding contracts, a (conditional) commitment corresponds to a (conditional) offer. A commitment can be in one of five states: act (offer in force), exp (offer expired), bas (base commitment; offer was taken up), sat (satisfied as the condition is brought about), and vio (violated as the condition cannot be brought about). Based on the above, some apparent states are impossible: $\mathsf{CC}(x, y, p, q)$ cannot be bas or vio for any $p (\not\equiv \mathsf{T})$, $q$; $\mathsf{CC}(x, y, \mathsf{T}, q)$ cannot be act or exp for any $q$. We abstract out any deadlines associated with commitments and assume timeouts are *exogenous*, meaning not controlled by the agents. Figure 2 shows a simplified life cycle of a commitment, loosely based on previous works [4, 7]. Operations cause commitment states to change.

Consider a scenario where a buyer and a seller are exchanging goods for payment. A conditional commitment $\mathsf{CC}(buyer, seller, goods, payment)$ : act, established via *create*, denotes an obligation from the buyer to the seller that if the goods are delivered, the buyer will pay. In the event that the precondition $goods$ holds, the conditional commitment is *detached* and changes to a base commitment $\mathsf{CC}(buyer, seller, \mathsf{T}, payment)$ : bas. In the event that $payment$ holds, the buyer's commitment is *discharged*. Commitments do not imply temporal ordering between their conditions and preconditions. For example, $payment$ may happen before $goods$, thus discharging the above conditional commitment.

Previous works describe the formal semantics of the commitment operations, especially in the face of concurrency [4]. Other considerations include whether a debtor eschews all responsibility by delegating a commitment. Business scenarios can differ in this regard. This paper's examples involve retaining responsibility.

## 2.2 States and Transitions

The enactment of contracts, as specified via protocols, can be captured as a transition system. The states of the transition system consist of fluents; the transitions are labeled with the actions of the agents. Actions may cause the value of the fluents to change, thereby changing the state. By default, each action $p$ causes a fluent $p$ to hold in the resulting state. Thus, in the following, $p$ always refers to a fluent caused by an action but not to the action itself. The negation of $p$ is written $\overline{p}$; exactly one of $p$ and $\overline{p}$ holds in each state. Commitments and commitment conditions are fluents and commitment operations are actions represented by corresponding fluents. A '$\cdot$' denotes a precondition (which may or may not be $\mathsf{T}$). Agents assign values to the states of the world. The valuations of actions are captured by valuations of the corresponding fluents.

## 2.3 Valuations

Each agent values each state privately and independently of other agents. For simplicity, the following examples assume that money is valued as itself. However, our approach does not depend on such an assumption. The values to an agent $a$ of a state $S$ and a fluent $p$ are denoted by $v_a(S)$ and $v_a(p)$, respectively. We assume that the valuations do not change during contract enactment.

- If a fluent $p$ corresponds to an action performed by $a$, then $v_a(p)$ is the cost of performing the action. This valuation does not take into account the valuation that the agent has for other effects caused by the action. For example, if $p$ represents the fact that $a$ paid \$5, then the value to $a$ of performing $p$ is $-\$5$. An effect of paying \$5 may bring about other fluents besides $p$ that $a$ values independently of $v_a(p)$. If $p$ does not correspond to an action of $a$, $v_a(p)$ may be positive if $a$ benefits from $p$. For example, if $p$ is a fluent representing the fact that $a$ has received \$5 from another agent, then the value to $a$ of $p$ may be \$5.

- The value to $a$ of a fluent $\mathsf{CC}(a, b, \cdot, q)$ : sat does not take into account the values of the actions $a$ performs to satisfy the commitment. For example, the value to $a$ of $\mathsf{CC}(a, b, \mathsf{T}, \$5)$ : sat disregards the value of paying \$5, but may include an improvement in $a$'s reputation resulting from its satisfaction.

- The value to $a$ of a fluent $\mathsf{CC}(a, b, \mathsf{T}, q)$ : vio may be the penalty $a$ has to pay for violating the commitment.

- The value to $b$ of a fluent $\mathsf{CC}(a, b, \cdot, q)$ : sat does not take into account the value to $b$ of the condition $q$ that $a$ brings about to satisfy the commitment. For example, \$5 value of the condition is separate from $v_b(\mathsf{CC}(a, b, \mathsf{T}, \$5)$ : sat$)$.

- The value to $b$ of a fluent $\mathsf{CC}(a, b, \mathsf{T}, q)$ : vio does not take into account the missed value of the condition $q$ that $a$ fails to bring about. For example, $v_b(\mathsf{CC}(a, b, \mathsf{T}, \$5)$ : vio$)$ may be the compensation to $b$ paid by the legal context in which the commitment exists.

## 2.4 Basic Valuation Constraints

Given the above basic valuations, we adopt from Yolum & Singh [19], the following constraints on how agents may value the various states that may arise during enactment of business contracts. For a given business environment and a business contract enacted in it, only a subset of these value constraints may hold. Also, we assume that the agents satisfy or violate commitments entirely by choice and not because of other constraints. For example, even though an agent has no control over natural calamities, it may choose to allocate sufficient resources to satisfy its commitments. If it does not allocate sufficient resources, and the commitments are violated, that is interpreted as being the agent's choice.

*Performing an action always incurs a cost to the performing agent.* If a fluent $p$ corresponds to an action performed by $a$, then

$$v_a(p) < 0 \tag{1}$$

This does not take into account the valuation that the agent has for other effects of the action. This constraint rules out altruistic agents, who may derive positive value out of performing actions for others.

*Debtors value a base commitment higher than the deed, though both are negative.*

$$v_a(q) < v_a(\mathsf{CC}(a, b, \mathsf{T}, q) : \mathsf{bas}) < 0 \tag{2}$$

Since promising to perform an action is better than performing the action itself, the debtor prefers the former over the latter. Since bringing about the condition $q$ always incurs a cost to the debtor, for the debtor, a base commitment is worse than having no commitment for $q$.

*Creditors assign positive value to the condition of the commitment.* If $\mathsf{CC}(a, b, p, q) : \mathsf{act}$, then

$$v_b(q) > 0 \tag{3}$$

That is, commitments are always favorable to creditors.

*Creditors prefer the deed over a base commitment to perform the deed, and both are positive.*

$$v_b(q) > v_b(\mathsf{CC}(a, b, \mathsf{T}, q) : \mathsf{bas}) > 0 \tag{4}$$

This captures the intuition that since debtors may choose to violate their commitments, creditors prefer to have the condition brought about over having a base commitment. Also, being creditor of a base commitment is better than being creditor of no commitment because, with a commitment, the prospect of satisfying the commitment remains alive.

*Valuation distributes over conjunction of fluents.*

$$v(p \wedge q) = v(p) + v(q) \tag{5}$$

This constraint rules out combinatorial and substitutional valuations. Combinatorial valuations apply when the value of a combination of items is greater than the sum of the value of individual items. Substitutional valuations apply when the value of a combination of items is less than the sum of the values of the individual items.

# 3. PROPOSED TECHNICAL FRAMEWORK

This section introduces the enhancements to the above that enable us to formalize and develop algorithms for contract checking.

## 3.1 Advanced Valuation Constraints

*Debtors derive positive value by satisfying commitments.*

$$v_a(\mathsf{CC}(a, b, \cdot, q) : \mathsf{sat}) > 0 \tag{6}$$

This rules out agents who do not prefer to keep their commitments. It also rules out environments where keeping commitments does not increase the reputation of the agent.

*For debtors, the benefit of satisfying commitments does not offset the cost of bringing about the condition.*

$$v_a(\mathsf{CC}(a, b, \cdot, q) : \mathsf{sat}) + v_a(q) < 0 \tag{7}$$

This rules out environments wherein a debtor values reputation gain above the cost of discharging the corresponding commitment.

*For debtors, the penalty of violation is worse than the cost of discharging a commitment.*

$$v_a(\mathsf{CC}(a, b, \mathsf{T}, q) : \mathsf{vio}) < v_a(q) + v_a(\mathsf{CC}(a, b, \mathsf{T}, q) : \mathsf{sat}) \tag{8}$$

This rules out environments in which the violation of a commitment may be a better choice than satisfying the commitment. Thus, unlawful agents who prefer to violate their commitments are ruled out. Also, lawless business environments where punishments for violators are nonexistent or lenient are ruled out. This constraint holds for multiple commitments: agents may still choose to violate low-priority commitments to ensure satisfaction of high-priority commitments, when both cannot be satisfied.

*Debtors create commitments that are beneficial to them.* For an agent $a$ if $\mathsf{CC}(a, b, p, q) : \mathsf{act}$, then

$$v_a(p) + v_a(q) + v_a(\mathsf{CC}(a, b, \cdot, q) : \mathsf{sat}) > 0 \tag{9}$$

This rules out irrational agents who create commitments that may not benefit them. In the case of nested commitments, this applies only to the outermost commitment: the inner commitments may not all be beneficial individually, but the outer commitment as a whole must be beneficial. A corollary of this constraint is that debtors prefer creating conditional commitments over inaction. However, the value of an active commitment is bounded by the benefit from the trade corresponding to the commitment.

$$v_a(p) + v_a(q) + v_a(\mathsf{CC}(a, b, \cdot, q) : \mathsf{sat}) > v_a(\mathsf{CC}(a, b, p, q) : \mathsf{act}) > 0 \tag{10}$$

*Creditors assign no value to a discharged commitment beyond the value of the condition.*

$$v_b(\mathsf{CC}(a, b, \cdot, q) : \mathsf{sat}) = 0 \tag{11}$$

*Creditors assign no value to a violated commitment beyond any compensation from the context.*

$$v_b(\mathsf{CC}(a, b, \mathsf{T}, q) : \mathsf{vio}) = 0 \tag{12}$$

*Debtors and creditors assign no value to an expired commitment.*

$$v_a(\mathsf{CC}(a, b, p, q) : \mathsf{exp}) = 0 \tag{13}$$

$$v_b(\mathsf{CC}(a, b, p, q) : \mathsf{exp}) = 0 \tag{14}$$

An expired conditional commitment is akin to an expired offer and has the same value as there being no commitment.

## 3.2 Models of Valuation Constraints

The above constraints reflect our intuitions about valuations of commitments and states by rational agents. We now show that the above constraints have models and thus are consistent. We also show via an example that these models promote trade when there are gains to be made by the parties. The following section illustrates the importance of commitments for encouraging cooperation in business environments.

A *protocol* enacts a contract. In the following, a *strategy* of an agent determines its choice of actions at each state of the protocol enacting the contract. An *outcome* of a strategy is a state in which the protocol may terminate if the agent follows the strategy. A strategy is *dominant* for an agent if the agent values all possible outcomes of the strategy higher than the possible outcomes of alternative strategies, regardless of the strategies of other agents. A *pure-strategy Nash equilibrium* is a set of deterministic strategies, one for each agent, such that no agent can garner higher value by unilaterally changing its current strategy.

### *3.2.1 Commitments and Rationality*

As an example, let us assume that a buyer and a seller have agreed to trade goods for money. Also, both the buyer and the seller are rational and have the valuations as shown in Table 3.2.1. These valuations are used in the following for showing that (1)–(14) have a model and for computing the pure-strategy Nash equilibria. However, like (1)–(14), the preferences (15) and (18)–(19) originate from intuition and not from the valuations of Table 3.2.1. Thus, verifying that the valuations of Table 3.2.1 is a model of (1)–(14), (15), and (18)–(19) means that the intuitions are consistent. For brevity, in the following, $g$, $p$, $C_{sb}$, and $C_{bs}$ denote *goods*, *pay*, $\mathsf{CC}(S, B, pay, goods)$, $\mathsf{CC}(B, S, goods, pay)$, respectively.

| | $\mathbf{g}$ | $\overline{\mathbf{g}}$ | $\mathbf{p}$ | $\overline{\mathbf{p}}$ |
|---|---|---|---|---|
| Seller | $-4$ | 0 | 5 | 0 |
| Buyer | 6 | 0 | $-5$ | 0 |

| | $\mathbf{C_{sb}}$ | | | | $\mathbf{C_{bs}}$ | | | |
|---|---|---|---|---|---|---|---|---|
| | act | bas | sat | vio | act | bas | sat | vio |
| Seller | 0.5 | $-3$ | 2 | $-3$ | 0 | 3 | 0 | 0 |
| Buyer | 0 | 4 | 0 | 0 | 0.5 | $-4$ | 2 | $-4$ |

**Table 1: Valuations of the buyer and the seller in purchase**

According to these valuations, like in usual practice, a trade is beneficial to both parties. First, let us assume an environment where commitments are not enforced. This is typical of open environments without any regulating agency and corresponds to assumptions commonly made in game-theoretic approaches. In this environment, the only possible states are $S_1 = \{\overline{g} \wedge \overline{p}\}$, $S_2 = \{\overline{g} \wedge p\}$, $S_3 = \{g \wedge \overline{p}\}$, and $S_4 = \{g \wedge p\}$. Because the seller is rational, it values these states with the following relationships.

$$v_s(S_2) > v_s(S_4) > v_s(S_1) > v_s(S_3) \tag{15}$$

The best outcome for the seller is $S_2$: it receives payment but doesn't send the goods. If commitments are not enforced, the seller would try to achieve $S_2$. Similarly, the buyer would prefer receiving goods and not sending a payment.

By applying (5) to (15), and canceling out terms on both sides, we obtain the following inequalities.

$$v_s(\overline{g}) \quad > \quad v_s(g) \tag{16}$$

$$v_s(g) + v_s(p) \quad > \quad v_s(\overline{g}) + v_s(\overline{p}) \tag{17}$$

Table 2 shows, in each cell, the valuation of the seller followed by that of the buyer. It shows the pure-strategy Nash equilibrium in bold. This means that with the valuations of Table 3.2.1, a rational buyer and seller would not trade.

| | Send payment | | Do not send payment | |
|---|---|---|---|---|
| Send goods | 1 | 1 | $-4$ | 6 |
| Do not send goods | 5 | $-5$ | **0** | **0** |

**Table 2: Pure-strategy Nash equilibrium without commitments**

Now, let us assume a commitment $C_{sb} = \mathsf{CC}(S, B, pay, goods)$ exists between the buyer (B) and the seller (S). Hence, the seller has committed to sending the goods if payment is received. Also, Constraints (1)–(14) are in force. For brevity, let $g$ denote *goods* and let $p$ denote *pay*. Then, the only possible states are:

$S_5 = \{C_{sb} : \mathsf{act} \wedge \overline{g} \wedge \overline{p}\}$ $\quad S_6 = \{C_{sb} : \mathsf{bas} \wedge \overline{g} \wedge p\}$
$S_7 = \{C_{sb} : \mathsf{sat} \wedge g \wedge \overline{p}\}$ $\quad S_8 = \{C_{sb} : \mathsf{sat} \wedge g \wedge p\}$
$S_9 = \{C_{sb} : \mathsf{vio} \wedge \overline{g} \wedge p\}$

For the seller to send goods, and still benefit from the contract, it should value these states with the following inequalities while complying with the constraints in force.

$$v_s(S_8) > v_s(S_5) > v_s(S_7) \tag{18}$$

$$v_s(S_8) > v_s(S_9) \tag{19}$$

The seller's valuations reflect the following intuitive preferences: commitment satisfaction above commitment violation, due to (19); trade better than no trade; and no trade better than trading without a commitment, due to (18). Both (19) and (18) can be inferred from (1)–(14), implying that no additional constraints are needed to motivate the seller to cooperate.

### *3.2.2 Consistency of Constraints*

Now let us show that (1)–(14) are consistent with (15) and (18)–(19). Doing so will prove that there are models of the valuation constraints such that the agents can discharge their commitments and still benefit individually.

By applying (5) to (18)–(19), and canceling out terms, we obtain the following inequalities.

$$v_s((C_{sb} : \mathsf{bas}) + p) \quad > \quad v_s((C_{sb} : \mathsf{act}) + \overline{p}) \tag{20}$$

$$v_s((C_{sb} : \mathsf{act}) + \overline{g}) \quad > \quad v_s((C_{sb} : \mathsf{sat}) + g) \tag{21}$$

$$v_s((C_{sb} : \mathsf{sat}) + g + p) \quad > \quad v_s((C_{sb} : \mathsf{vio}) + \overline{g} + p) \tag{22}$$

To prove that (16)–(17) and (20)–(22) along with (1)–(14) are consistent, demonstrating a model is sufficient. It is easy to verify that the valuations of Table 3.2.1 satisfy all of the above constraints and, thus, Table 3.2.1 is a model.

A similar result can be obtained from the perspective of the buyer. The result trivially extends to generalized commitments, the trading example discussed here is merely an illustration. However, as shown in Table 3, the pure-strategy Nash equilibrium with just one commitment does not promote trade: the commitment motivates the seller but not the buyer. Note that the pure-strategy Nash equilibrium is not a dominant strategy for the seller, but is for the buyer.

| | Send payment | | Do not send payment | |
|---|---|---|---|---|
| Send goods | 3 | 1 | $-2$ | 6 |
| Do not send goods | 2 | $-5$ | **0** | **0** |

**Table 3: Pure-strategy Nash equilibrium with $C_{sb}$**

Let us assume that the buyer and the seller both have commitments to each other: $C_{bs} = \mathsf{CC}(B, S, goods, pay)$ and $C_{sb} = \mathsf{CC}(S, B, pay, goods)$. Constraints (1)–(14) along with (16)–(17) and (20)–(22) are in force. Thus, the only possible states are:
$S_{10} = \{C_{sb} : \mathsf{act} \wedge C_{bs} : \mathsf{act} \wedge \overline{p} \wedge \overline{g}\}$

$S_{11} = \{C_{sb} : \mathsf{bas} \wedge C_{bs} : \mathsf{sat} \wedge p \wedge \overline{g}\}$
$S_{12} = \{C_{sb} : \mathsf{sat} \wedge C_{bs} : \mathsf{bas} \wedge \overline{p} \wedge g\}$
$S_{13} = \{C_{sb} : \mathsf{sat} \wedge C_{bs} : \mathsf{sat} \wedge p \wedge g\}$
$S_{14} = \{C_{sb} : \mathsf{sat} \wedge C_{bs} : \mathsf{vio} \wedge \overline{p} \wedge g\}$
$S_{15} = \{C_{sb} : \mathsf{vio} \wedge C_{bs} : \mathsf{sat} \wedge p \wedge \overline{g}\}$

Given the valuations of Table 3.2.1, the strategy to trade (send goods and send payment, respectively) is the dominant strategy and is one of the pure-strategy Nash equilibria as shown in Table 4. Its dominance means that no additional constraints need to be enforced for motivating cooperation.

| | Send payment | | Do not send payment | |
|---|---|---|---|---|
| Send goods | **3** | **3** | $-2$ | 2 |
| Do not send goods | 2 | $-3$ | **0** | **0** |

**Table 4: Pure-strategy Nash equilibria with $C_{sb}$ and $C_{bs}$**

## 3.3 Contract Correctness

This section defines basic terminology and describes some interesting correctness properties for contracts via examples.

A contract $\mathcal{C}$ is represented as a set of commitments. For example, the contract between a buyer and a seller engaged in the purchasing can be represented as $\mathcal{C}$=$\{\mathsf{CC}(S, B, pay, goods)$, $\mathsf{CC}(B, S, goods, pay)\}$.

DEFINITION 1. *A protocol is a specification of a set of coordination constraints on the actions of the agents.* ▮

We specify two kinds of coordination constraints. A precedence constraint $a \prec b$ means that $a$ must occur before $b$. A mutual exclusion constraint $a$ XOR $b$ means that exactly one of $a$ or $b$ must occur. For example, the protocol for the above example may be specified as $\mathbb{P}$=$\{pay \prec goods\}$.

DEFINITION 2. *An agent is rational if it always chooses a course of action that leads to the most beneficial states in the foreseeable future.* ▮

Informally, the foreseeable future is the set of future states that are known to the agents making the choice. In our formulation, the protocol yields the known future states. A rational agent would choose the course of action depending on whether or not it knows the other agents to be rational. If all agents know each other to be rational, and know that each of them knows that, and so on, then the agents are *publicly* rational. We only consider publicly rational agents in the following.

DEFINITION 3. *A contract is* rationally beneficially omni-correct *(rbo) with respect to a set of value constraints if it guarantees that all agents benefit from participating in it as long as the specified constraints hold, regardless of the protocol.* ▮

For example, with (1)–(14), in the contract $\mathcal{C}$=$\{\mathsf{CC}(S, B, pay, goods)$, $\mathsf{CC}(B, S, goods, pay)\}$, both the buyer and the seller benefit regardless of the temporal order of $pay$ and $goods$.

For practical significance, it is more useful to define correctness from the perspective of individual agents. A contract is correct from the perspective of an agent if the agent benefits from it.

DEFINITION 4. *A contract is* rationally beneficially uni-correct *(rbu) from the perspective of an agent with respect to a set of value constraints if it ensures that the agent benefits from participating in it as long as the specified constraints hold.* ▮

Thus, rbo-correctness is equivalent to rbu-correctness for each agent. Consequently, we need only an algorithm for checking rbu-correctness: rbo-correctness can be inferred from it. Also, rbo-correctness implies rbu-correctness for each agent. Guaranteed benefit may be unnecessarily strict. It may be sufficient to check if an agent would not incur losses by participating in the contract.

DEFINITION 5. *A contract is* rationally safely omni-correct *(rso) with respect to a set of value constraints if it ensures that no agent incurs losses from participating in it as long as the specified constraints hold, regardless of the protocol.* ▮

As for benefit, safety from the perspective of a rational agent can be defined as rsu-correctness. Thus, rso-correctness is equivalent to rsu-correctness for each agent. For example, $\mathcal{C}$=$\{\mathsf{CC}(x, y, p)\}$ is rsu-correct for $y$ relative to the valuations $\mathcal{V}$=$\{v_y(p) > 0,\ v_x(p) < 0\}$. However, $\mathcal{C}$ is not rsu-correct for $x$ as it can only lose by participating in this contract.

DEFINITION 6. *A contract is* rationally beneficially omni-correct *under a protocol (rbop) with respect to a set of value constraints and a protocol if it ensures that all agents benefit from participating in it as long as the specified constraints hold and the specified protocol is followed.* ▮

For example, in the contract $\mathcal{C}$=$\{\mathsf{CC}(B, S, pay, goods)\}$, given (1)–(14), both the buyer and the seller benefit as long as the protocol $\mathbb{P}$=$\{pay \prec goods\}$ is followed. Absent the protocol, the seller may deliver the goods, thus discharging $\mathsf{CC}(B, S, pay, goods)$ but the buyer will not pay if $v_B(pay) < 0$ (and if it is rational). Thus, this contract would be harmful to the seller.

Similarly, rbup-correctness can be defined from the perspective of a rational agent. In the above example, if the protocol does not constrain the order of the seller's and buyer's actions, the contract is not rbup-correct for the seller. Protocol-based correctness too can be considered relative to safety instead of benefit. Thus, we can define rsup-correctness and rsop-correctness.

This paper presents algorithms for checking rsup and rbup correctness. However, rsop and rbop correctness can be inferred from rsup and rbup correctness for every agent, respectively. Also, rsu and rbu correctness are special cases of rsup and rbup correctness with no coordination constraints, respectively. Thus, the algorithms presented in the following can also be used to check rsu, rbu, rso, and rbo correctness of contracts.

Other interesting properties include correctness relative to partial knowledge of constraints and knowledge of the rationality of only a subset of agents. We defer these cases as well as correctness from the perspective of an irrational agent to future study.

# 4. CHECKING CORRECTNESS

This section describes (1) our overall methodology and tools for semiautomatically checking two kinds of contract correctness properties and (2) algorithms for checking these correctness properties.

## 4.1 Methodology

Figure 3 depicts the steps of our methodology. The dashed edges denote manual steps whereas the solid edges denote steps automated by tools. A contract designer specifies and translates a legal contract into commitments (steps 1 and 2). Methodologies similar to those of Milosevic *et al.* [11] can be adapted for commitments. The contract designer also specifies the protocol coordination constraints (step 3).

We employ the causal logic $C+$ [8] to declaratively specify the protocols and commitments and the tool CCalc [16] for generating

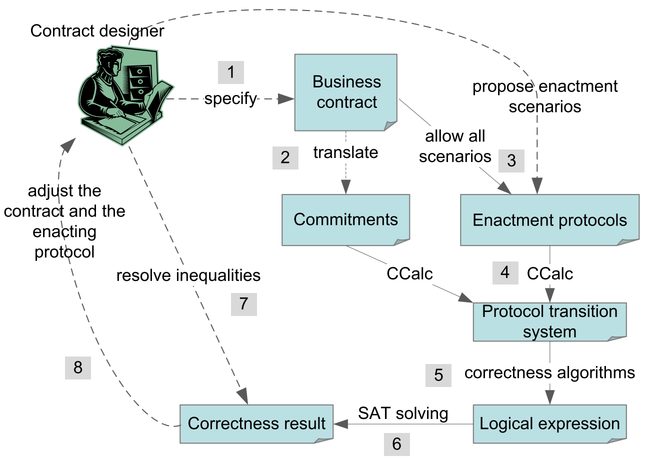

**Figure 3: A methodology for checking correctness of contracts**

a transition system for the specified protocol (step 4). Our algorithms operate on a transition system and output a Boolean formula of inequality constraints on the valuations of various protocol states (step 5). The Boolean formula is then evaluated via a SAT solver using the inequalities that hold based on the constraints of Sections 2.4 and 3.1 (step 6). However, not all inequalities can necessarily be resolved by those constraints. This is because the states may contain multiple commitments such that they cannot be ranked according to the agent's preferences. Also, a business partner's preferences may not be known. Either the contract designer decides on the truth of such inequalities (step 7) or the solver assumes them to be false. The truth or falsity of the overall formula reflects whether or not the contract possesses the concerned property. Depending on the correctness result, the contact designer may adjust the terms of the contract or the coordination constraints (step 8). We have prototyped the above tools for such semiautomatic correctness checking of formally specified contracts and protocols.

## 4.2 Algorithms

Given the constraints imposed on the preferences of the agents due to rationality and the definitions of the various kinds of correctness properties, we need algorithms to check whether a given contract and protocol possess these properties. This section presents an algorithm to check rbup-correctness and describes how an algorithm for rsup-correctness can be derived from it. As described above, these algorithms can also be used to compute or infer other correctness properties.

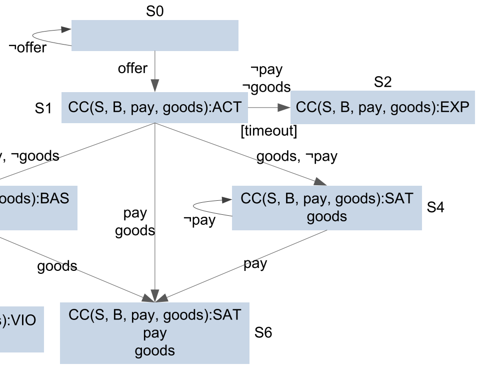

**Figure 4: Transition system for a purchase protocol**

We illustrate the algorithms with a purchase protocol that adds an offer message to the above pay-goods example. The seller (S) makes an offer that creates the commitment. Payment and goods can be exchanged in any order. Thus, the only coordination constraints are $\mathbb{P}$={*offer* $\prec$ *offerpay*, *offer* $\prec$ *offergoods*}. Figure 4 shows the corresponding transition system generated via CCalc.

From each state, multiple agents may act simultaneously. Thus, each transition may have multiple actions, one for each agent. Transitions are labeled with conjunctions of action literals. Action literals can be positive or negative, respectively denoting the occurrence or nonoccurence of an action. A transition can have at most one action literal from an agent. If protocol constraints allow only the nonoccurrence of an action from a state, then its literal is discarded from the transition label. A timeout may cause a transition that is labeled with only negative literals and is not a self-loop. For example, $s_1 \longrightarrow s_2$ is a timeout transition.

A protocol state is *transient* if it has at least one outgoing timeout transition. For example, $s_1$ is a transient state in Figure 4. An agent $x$ *controls* a transition $t_i$ via an action $\alpha$, denoted by *controls(*$x$*,* $t_i$*,* $\alpha$*)*, if $\alpha$ is an action of $x$ and is required for the transition. An agent $x$ *controls* a state $s_i$ via $\alpha$, denoted by *controls(*$x$*,* $s_i$*,* $\alpha$*)*, if for at least one of the outgoing transitions $t_i$ of the state, *controls(*$x$*,* $t_i$*,* $\alpha$*)* holds. For example, the seller and the buyer both control $s_1$. A transition $t_i$ from $s_i$ to $s_j$ is denoted by $s_i \overset{t_i}{\longrightarrow} s_j$ and a path from $s_i$ to $s_j$ is denoted by $s_i \leadsto s_j$.

A state has a timeout transition for each commitment whose state is either act or bas. Upon timeout, act is replaced by exp and bas is replaced by vio.

A state $s_i$ is a *terminal* state, denoted by *terminal(*$s_i$*)*, if it is not the start state and not a transient state such that: (a) it has no outgoing transitions or (b) all agents that control $s_i$ would choose to stay in that state. If (a), then the condition for the state being terminal, denoted by *cond(terminal(*$s_i$*))*, is true. Otherwise, the condition reflects whether or not all agents choose to stay in the state. For example, $s_4$ is a terminal state as long as $v_b(s_4) > v_b(s_6)$.

An action $\beta$ is an *alternative* to an action $\alpha$ of $x$ at state $s_i$, denoted by *alt(*$\beta, \alpha, s_i, x$*)*, if $\beta$ and $\alpha$ are different named actions or $\beta$ and $\alpha$ have opposite polarity. For example, an alternative action of *pay* at $s_1$ for the buyer is $\neg$*pay*.

An action $\alpha$ *requires* an action $\gamma$ at state $s_i$, denoted by *req(*$\alpha$*,* $\gamma$*,* $s_i$*)*, if on all transitions from $s_i$ where $\alpha$ occurs, $\gamma$ also occurs. The purchase example does not demonstrate this relationship.

In the following, $\mathbb{A}$ is the set of all (rational) agents, $\Omega$ the set of all actions, and $\mathbb{T}$ the set of all transitions in the given protocol.

Algorithm 1 presents *choice()* that returns an expression that reflects whether or not $x$ will choose $\alpha$ at $s_i$. Intuitively, $x$ will choose $\alpha$ if it is better than all other alternatives $\beta$ of $\alpha$.

**Algorithm 1**: choice($x$, $s_i$, $\alpha$): Check if $x$ will choose $\alpha$ at $s_i$

```
conj ← TRUE;
foreach $\beta \in \Omega$:$alt(\beta,\alpha,s_i,x)$ do
    conj ← conj · "∧" · better($x$,$s_i$,$\alpha$,$\beta$);
return conj;
```

Algorithm 2 presents *better()* that returns a logical expression that reflects whether $\alpha$ is better than $\beta$ for $x$ at $s_i$. This means that for all transitions that $x$ controls via $\alpha$ at $s_i$ (a) all required actions are chosen by the concerned agents (lines 3–4), and either (b) if the transition is to a terminal state $s_j$, then for all the terminal states $s_\beta$ reachable from $s_i$ via $\beta$: either (1) $s_j$ is better than $s_\beta$, or (2) the path to $s_\beta$ will not be chosen, or (3) the condition for either $s_j$ or $s_\beta$ being terminal does not hold (lines 5–8), or (c) if $s_j$ is not a terminal state then, at least one action will be chosen at $s_j$.

**Algorithm 2**: better($x$, $s_i$, $\alpha$, $\beta$): Check if for $x$ $\alpha$ is better than $\beta$ at $s_i$

```
1  conj ← TRUE;
2  foreach $t_i \in \mathbb{T}$ at $s_i$: controls($x$,$t_i$,$\alpha$) do
3      foreach $y \in \mathbb{A}, y \neq x$: controls($y$,$t_i$,$\gamma$) $\wedge$ req($\alpha$,$\gamma$,$s_i$) do
4          conj ← conj · "$\wedge$" · choice($y$,$s_i$,$\gamma$);
5      if $s_i \xrightarrow{t_i} s_j$:terminal($s_j$) $\vee$ $s_i = s_j$ then
6          foreach $s_\beta \neq s_j$: terminal($s_\beta$) $\wedge$ $s_i \rightsquigarrow s_\beta$ do
7              expr ←
               "$(v_x(s_j) > v_x(s_\beta) \vee \neg cond(terminal(s_j)) \vee \neg cond(terminal(s_\beta)) \vee \neg path(s_i \rightsquigarrow s_j))$";
8              conj ← conj · "$\wedge$" · expr;
9      else
10         expr ← FALSE;
11         foreach $z \in \mathbb{A}$:controls($z$,$s_j$,$\delta$) do
12             expr ← expr · "$\vee$ (" · conj · "$\wedge$" choice($z$, $s_j$, $\delta$) ")";
13         conj ← expr;
14 return conj;
```

Verify that neither the expression returned by *choice(s, $s_0$, offer)* nor the expression returned by *choice(s, $s_0$, $\neg$offer)* can be satisfied. When neither expression can be satisfied, this implies that there is no dominant pure-strategy Nash equilibrium for the given agent (as Section 3.2 discusses for the one-commitment case).

PROPOSITION 1. [Soundness] *From a start state, if* choice($x$, $s_0$, $\alpha$) *returns a satisfiable expression, and the valuations are consistent with (1)–(14), there exists a pure-strategy Nash equilibrium.*
***Proof.*** *Say* choice($x$, $s_0$, $\alpha$) *returns a satisfiable expression. This implies that one of the agents $x$ has a choice from $s_0$ that ensures that for all possible terminal states $s_j$ from $s_0$, $x$ is better off than all possible terminal states $s_\beta$ possible by doing $\beta$ from $s_0$ (line 7 in Algorithm 2). This implies that $x$ will always unilaterally choose $\alpha$ from $s_0$. Also, in subsequent states, on the path to the chosen terminal states, other agents must have chosen at least one action (lines 11–13 in Algorithm 2). Say a Nash equilibrium does not exist. This implies that for at least one agent, there is a strategy better than its current choice. Also, from the resulting set of chosen strategies (one per agent), again, there is another agent that can benefit by changing its choice, and so on. But, for $x$ and the other agents with their chosen actions, there are no better alternatives (lines 2–3 in Algorithm 1). Thus, no agent would change its strategy. Thus, an equilibrium must exist.* ▮

PROPOSITION 2. [Incompleteness] *From a start state, although* choice($x$, $s_0$, $\alpha$) *does not return a satisfiable expression, and the valuations are consistent with (1)–(14), there may exist a pure-strategy Nash equilibrium.*
***Proof.*** *To prove incompleteness, a case where* choice() *returns an unsatisfiable expression even though a Nash equilibrium exists is sufficient. Figure 5 shows such a case. Say $x$ controls $\alpha$ and $y$ controls $\beta$. Also, $v_x(\beta) > 0$ and $v_y(\alpha) > 0$. There are no commitments between $x$ and $y$. However, the valuations $v_x(s_8) > v_x(s_{10})$ $> v_x(s_7) > v_x(s_9)$ are consistent with (1)–(14). The Nash equilibrium is for both the agents to not act. However,* choice() *for all actions from $s_7$ would return unsatisfiable expressions.* ▮

Algorithm 3 presents a method to check rbup-correctness of contracts. If an expression returned by *rbup-correct($s_i$, $s_r$, $x$)* can be satisfied, it means that the contract guarantees that the protocol terminates in a state better than $s_r$ for $x$ starting at $s_i$. A call to *rbup-correct($s_0$, $s_0$, $x$)* would check the same from the starting state $s_0$. Algorithm 3 checks that for all agents $z$ (including $x$) that may act from $s_i$, and make a choice to do $\gamma$, all terminal states $s_j$ possible as a result of $\gamma$ at $s_i$ are beneficial to $x$, i.e., $(v_x(s_j) > v_x(s_r))$.

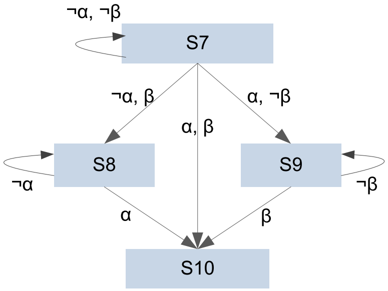

**Figure 5: A protocol for proving incompleteness of *choice***

**Algorithm 3**: rbup-correct($s_i$, $s_r$, $x$): Check rbup-correctness relative to $s_r$ from the perspective of $x$ starting at state $s_i$

```
1  conj ← TRUE;
2  foreach $z \in \mathbb{A}$, $\gamma \in \Omega$:controls($z$,$s_i$,$\gamma$) do
3      conj ← conj · "$\wedge$" · choice($z$,$s_i$,$\gamma$);
4      foreach $t_i \in \mathbb{T}$:controls($z$,$t_i$,$\gamma$) do
5          if $s_i \xrightarrow{t_i} s_j$:terminal($s_j$) $\vee$ $s_i = s_j$ then
6              expr ←
               "$(v_x(s_j) > v_x(s_r) \vee \neg cond(terminal(s_j)))$";
7              conj ← conj · "$\wedge$" · expr;
8          else
9              conj ← conj · "$\wedge$" · rbup-correct($s_j$, $s_r$, $x$);
10 return conj;
```

The algorithm for checking rsup-correctness is not presented here. This is because the only difference between safety and benefit is that instead of checking $(v_x(s_j) > v_x(s_r))$ as in rbup-correct (line 6), we would check for $(v_x(s_j) \geq v_x(s_r))$ in rsup-correct.

PROPOSITION 3. [Safety] rsup-correct($s_0$, $s_0$, $x$) *returns a satisfiable expression if the valuations are consistent with (1)–(14).*
***Proof.*** *If $x$ has no choice from $s_0$ that is safe, $x$ will not act, remain in $s_0$, and be safe as $v_x(s_0) \geq v_x(s_0)$. If $x$ does not control $s_0$, as a creditor it can let the unsafe commitments created by the other agents expire, and still be safe. Due to (9), $x$ will not act to create a commitment that does not benefit it as a debtor.* ▮

However, an algorithm for rsup-correctness is useful because it enables a designer to explore if progress can be made safely and how knowledge of the truth or falsity of a subset of the inequality constraints affects the outcome.

## 5. DISCUSSION

This paper addresses the important problem of reasoning about the correctness of business contracts. Generic correctness checking of business contracts via low-level representation (such as finite state machines) has only limited business applicability. Instead, we focus on the *value* that contract execution would bring to the participants. We present a valuation model based on earlier work on commitments, by representing a contract as a collection of commitments, with values associated with satisfying or violating commitments. We define correctness properties for contracts and present a methodology and algorithms for verifying them.

## 5.1 Literature

Checking business contracts for correctness is an active research area. Molina-Jiménez *et al.* [12] present a technique for checking contract correctness by modeling the contract as a finite state machine (FSM). They map the rights and obligations extracted from the clauses of the contract into the states, transition and output functions, and input and output symbols of a FSM. Molina-Jiménez *et al.* can verify that the clauses stipulated in the contract are observed when the contract is executed. However, they verify the correctness of a contract as stated. By contrast, we investigate the different variants by which the contract can be executed, and their relative values to participating agents. In a similar vein, Wan & Singh [17] present a method to verify a collection of multiparty commitments for correctness, and produce executions under which progress takes place. However, the notion of correctness there is limited to preventing deadlock, and ignores the valuations of agents. Our basic valuation constraints were originally proposed by Yolum & Singh [19]. However, their emphasis is on rules via which agents make increasingly strong commitments toward each other to arrive at an agreement while minimizing the risks.

Governatori *et al.* [9] discuss the compliance of business processes with business contracts. They model the rights and obligations of participating agents, and present general guidelines for translating deontic logic expressions into business process tasks expressed in a language such as BPMN [13]. Governatori *et al.* seek to ensure that a business process adheres to a contract. By contrast, we check the correctness of contracts.

Radha Krishna *et al.*'s contract modeling framework $ER^{EC}$ formally represents a business contract along with the data and process models that realize it [10]. Although $ER^{EC}$ does not focus on contract correctness, it would be worthwhile to investigate how it can be combined with our commitment-based approach.

Our protocol scenario generation approach is inspired by Chopra & Singh's work on contextualizing protocols [3]. They provide examples of contextualizations such as, (for purchase) return-refund, reminder, and pay-before-goods. We introduced considerations of rationality and additional rules for eliminating infeasible scenarios. Expanding our protocol scenario generation to represent domain-specific customizations would be a fruitful direction.

Winikoff [18] presents a similar commitment-based interaction generation approach. However, it focuses only on designing agent interactions based on the commitments given. Bentahar *et al.* [2] present a formal semantics for a combined commitment and argumentation network. This may be used in our work to support argumentation. Argumentation may be needed for implementing mid-stream protocol adaptations due to exceptions.

The $e^3value$ project [5] is the closest to our work. It presents a model of the value exchanges that occur during business-to-business interactions. However, it lacks an account of correctness as developed here. In a similar vein, Pijpers & Gordijn [14] show how to derive business process models from value models. Baldoni *et al.* [1] present a language and approach for verifying conformance of a set of agent interactions to a defined protocol. The verification is based on modeling the protocol and interactions as finite-state automata, and comparing the two. Our work, by contrast, has a broader scope, since it views ensuring correctness from the contract perspective.

## 5.2 Future Work

Besides the possibilities discussed above, this topic opens up several avenues for further research, of which we list a few. One, we will augment our tools so that they produce explicit counterexamples, which can be used to refine a contract. Two, we will enhance our analysis technique to provide global contract correctness valuations and guarantees. Three, we will consider further problem scenarios so as to construct a comprehensive methodology.

# 6. REFERENCES

[1] M. Baldoni, C. Baroglio, A. Martelli, and V. Patti. Verification of protocol conformance and agent interoperability. *CLIMA VI*, pp. 265–283, 2005.

[2] J. Bentahar, B. Moulin, J.-J. C. Meyer, and B. Chaib-draa. A logical model for commitment and argument network for agent communication. *AAMAS*, pp. 792–799, 2004.

[3] A. K. Chopra and M. P. Singh. Contextualizing commitment protocols. *AAMAS*, pp. 1345–1352, 2006.

[4] N. Desai, A. K. Chopra, and M. P. Singh. Representing and reasoning about commitments in business processes. *AAAI*, pp. 1328–1333, 2007.

[5] $E^3$Value. 2007. http://www.e3value.com/.

[6] Foamex, AMFS, and Foamtec. Manufacturing agreement. http://contracts.onecle.com/admat/foamtec.mfg.1998.01.30.shtml.

[7] N. Fornara and M. Colombetti. Defining interaction protocols using a commitment-based agent communication language. *AAMAS*, pp. 520–527, 2003.

[8] E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, and H. Turner. Nonmonotonic causal theories. *Artificial Intelligence*, 153(1-2):49–104, 2004.

[9] G. Governatori, Z. Milosevic, and S. W. Sadiq. Compliance checking between business processes and business contracts. *EDOC*, pp. 221–232, 2006.

[10] P. R. Krishna, K. Karlapalem, and D. K. W. Chiu. An $\mathrm{ER}^{\mathrm{EC}}$ framework for e-contract modeling, enactment and monitoring. *Data & Know. Engg.*, 51(1):31–58, 2004.

[11] Z. Milosevic, S. W. Sadiq, and M. Orlowska. Translating business contracts into compliant business processes. *EDOC*, pp. 211–220, 2006.

[12] C. Molina-Jiménez, S. K. Shrivastava, E. Solaiman, and J. P. Warne. Run-time monitoring and enforcement of electronic contracts. *Elect. Comm. Res. Appl.*, 3(2):108–125, 2004.

[13] OMG. Business process modeling notation, 2007. http://bpmn.org/.

[14] V. Pijpers and J. Gordijn. Bridging business value models and process models in aviation value webs via possession rights. *HICSS*, p. 175, 2007.

[15] M. P. Singh. An ontology for commitments in multiagent systems. *Art. Intell. & Law*, 7:97–113, 1999.

[16] Texas Action Group at Austin. The causal calculator CCALC. http://www.cs.utexas.edu/users/tag/cc/.

[17] F. Wan and M. P. Singh. Formalizing and achieving multiparty agreements via commitments. *AAMAS*, pp. 770–777, 2005.

[18] M. Winikoff. Designing commitment-based agent interactions. *IAT*, pp. 363–370, 2006.

[19] P. Yolum and M. P. Singh. Enacting Protocols by Commitment Concession. *AAMAS*, pp. 116–123, 2007.