Business Rules for Compliant Business Process Models

Stijn Goedertier and Jan Vanthienen

Department of Decision Sciences & Information Management,
Katholieke Universiteit Leuven, Belgium
{myFirstName.myLastName}@econ.kuleuven.be

Abstract. The value of business rules in business process modeling has been recognized by several authors: if the knowledge about implicitly present policies and regulations is made explicitly traceable as business rules we can enhance the flexibility of computer-supported business processes. Yet to date it is not fully clear how business rules can be used to model compliant business process models. In this paper, we show that deontic assignments can be used to formally model business protocols. Within the regulatory framework of a business protocol, participants have the autonomy to perform activities according to their own business policies. This can be modeled using reaction rules.

1 Motivation and Methodology

Business Information Systems must flexibly support business processes that at the same time are compliant to the intra-organizational policies of the business and the inter-organizational regulations and protocols the business has to observe. Unfortunately, Information Systems have often made business processes more rigid than flexible, by slackening the business' ability to rapidly guarantee *business policy compliance* and *regulatory compliance*.

Flexibility is the ability to rapidly incorporate business policy change and regulatory change in the business processes. One aspect in obtaining flexibility of computer-supported business processes is to construct a common language between the business-side and IT-side of organizations. Such a language allows the business-side to formally represent models of how it operates internally and how it can legally interact with business partners. At the same time, such a common language allows the IT-side to have Information Systems support business processes accordingly, with as little development effort as possible. Ideally, Information System Technology must support business process models in such a way that they become human-understandable, yet machine-executable specifications.

The value of business rules in business process modeling has been recognized by several authors [1] [2] [3]. We define *business rules* as atomic, formal expressions of business policy and regulations that define or constrain some aspect of business. The key to business rules is that they make the knowledge about implicitly present policies and regulations explicitly traceable. In this way, changes in policies and regulations can be traced back to the business processes were they are to be enforced, thus enhancing flexibility.

Wagner classifies business rules into four categories: integrity constraints, derivation rules, reaction rules and deontic assignments [1]. While the role of integrity constraints, derivation rules and reaction rules in business processes descriptions is reasonably understood [4], it is less clear which role *deontic assignments* can play in the modeling of business processes.

In this paper, we define *deontic assignments* as the obligations, permissions and prohibitions of both internal and external partners in a business interaction. We show that deontic assignments can be used to formally model commitment-based business protocols, which can be syntactically verified, semantically validated and from which the public aspects of business process models can be generated for each of the business partners. Consequently companies can use the methodology for dealing with business policy and regulatory change, as is depicted in Fig. 1. Regulations are captured in terms of integrity constraints, derivation rules and deontic assignments. Regulations can be grouped to *business protocols*, which express the legally acceptable conduct of all process partners in a business interaction. In combination with *business policies*, which express the policy choices of the business and are formalized as integrity constraints, derivation rules and reaction rules, compliant business process models can be generated.

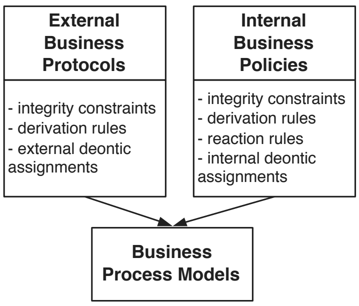

Fig. 1. Methodology: business rules for compliant business process models

The advantages of using business rules in business process modeling are clear. The traceability of the rules of business policies and business protocols enable companies to keep track of changes and to take into account dependencies between business rules. Moreover, because business protocols model business interactions from a third party perspective, the business protocols can become standard specifications in the business community, which can stimulate reuse or enable automatic verification of compliance. The rich semantics of the commitment-based business protocols, which are defined in terms of deontic assignments rather than in terms of mere sequence constraints, present new opportunities for verification and validation.

The remainder of the paper is structured as follows. In section 2 we outline the relevant literature on business rules and the modeling of business processes and commitment protocols. In section 3 we develop a lightweight metamodel of business protocols. In section 4 we ground the semantics of deontic assignments in formal logic. In section 5 we highlight some issues in the verification of business protocols. Finally, in section 6 we indicate how compliant business processes can be generated from business protocol specifications.

2 Related Work

Many authors recognize the value of business rules in requirements analysis [2] [5] [1] [6] [3], and software engineering [7]. As a consequence, many definitions of business rules exist in the literature, of which Wagner gives an overview [1]. We define *business rules* as *atomic, formal expressions of both business policy and business regulation that define or constrain some aspect of business.* With this definition we explicitly include business rules that originate from externally imposed regulations, because we do not want to rule out the possibility that certain business rules can be shared in a business community.

The Semantics of Business Vocabulary and Rules specification (SBVR), approved as a final adopted specification of the OMG [8], divides business rules into structural rules, which express necessities and supplement definitions, and operative rules, which express obligations and business conduct. Wagner categorizes business rules in four basic types [1]: integrity constraints, derivation rules, reaction rules and deontic assignemnts. **Integrity constraints** are assertions that must remain true and thus constraint the domain over which business facts can range. **Derivation rules** are statements that define new facts in terms of existing facts. **Reaction rules** state which actions are to be undertaken, given the occurrence of a certain business event. **Deontic assignments** are statements of the powers, rights and duties of internal agents. According to these definitions, it seems most plausible to relate integrity constraints and derivation rules to structural rules, whereas deontic assignments and reaction rules seemingly correspond to operative rules. Because Wagner's classification is more fine grained, we will adopt this classification in the rest of the paper.

Rules can play an important role in the modeling of business protocols and business processes. According to Bussler, business protocols describe the external, public behavior of agents in a business interaction, whereas business process models lay down the internal, private behavior [9]. This is supported by Peltz who states that business protocols describe business processes from a third-person perspective, rather than from a first-person perspective [10]. Accordingly we define *business process models* as *models of the long-running interactions between business partners, from the viewpoint of one of the business partners, who carries out both publicly visible and private activities in response to business events.* In contrast, *business protocols* are *models of the long-running interac-*

tions between business partners, from the perspective of an external observer, in which the private conduct is not modeled.

Deontic assignments can be used to model business protocols. Singh and Yolum have shown how deontic logic can be used in the modeling of commitment protocols [11] [12] [13]. Similar to commitments in commitment protocols, we give a broader interpretation to deontic assignments than Wagner. We define deontic assignments as *the obligations, permissions and prohibitions of both internal and external partners in a business interaction.* In this way deontic assignments can be used to express the legally accepted conduct of all process partners. Within the regulatory framework of a business protocol, participants have the autonomy to perform activities according to their own business policies. This can be modeled using reaction rules [4]. The use of business protocols for modeling business processes is different from traditional process modeling techniques such as Petri Nets because it deals with sequence constraints in a more semantic, natural way.

The concept of business protocols is closely related to that of business contracts. Kabilan describes an ontology-based methodology to design contract workflow models [14]. Grosof and Poon describe an approach to model business contracts using rules [15]. The approach in this paper is closer, however, to that of Singh and Yolum in that it makes use of reasoning mechanism over commitment-based business protocols which facilitate verification and automatic process model generation [16]. Rouached et al. describe an architecture for the event-based monitoring of business contracts [17]. Knottenbelt and Clark describe a language for intelligent agents to reason about the contract state [18]. The objective of this paper differs in that in does not consider the execution-time monitoring of business contracts, but rather considers the design-time modeling of compliant business processes.

The modeling of business protocols is relevant in the domain of web services and e-commerce, hence the specifications such as WS-Coordination, WS-Choreography or the RosettaNet Partner Interface Processes. A comprehensive view on web services is contained in the Web Service Modeling Framework (WSMF) [19]. Business rule-based business protocols could be one aspect in providing mediation of business logics. Business rules are also present in the Business Collaboration Development Framework (BCDF) of Orriëns et al. [20]. This framework strives for adaptability in business collaboration using development rules – which include business rules – for domain analysis , management rules for validation and verification and derivation rules for model transformation.

3 A Business Protocol Metamodel

It should be noted that the modeling of business protocols in terms of business rules is not limited to the commitments in business protocols, i.e. the control flow perspective, but also considers the data flow in business interactions. Figure 2 contains a part of the MOF/UML metamodel we use to represent business protocols.

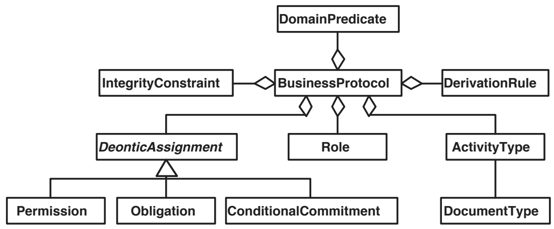

Fig. 2. A MOF/UML metamodel of a business protocol model

The data flow perspective in this metamodel is represented by modeling elements such as `DomainPredicate`, `IntegrityConstraint`, `DerivationRule` and `DocumentType`. Document types describe the types of documents that are used to communicate the performance of activities. As in the real world, documents are legal proof of the performance of activities of a certain activity type and are a legal, non-refutable acknowledgment of the rights of an opposite party that are attached to the legal act. For example, an order document is the legal proof that a buyer has placed an order. In addition, documents are the containers of data elements. Data elements are modeled using the `DomainPredicate` class. Domain predicates correspond to concepts such as role in Object Role Modeling [21] or fact types in SBVR [8]. At this level of abstraction, it suffices to know the domain predicates. It is unnecessary to determine how facts resort under objects. Integrity rules and derivation rules are used to constrain and define the business concepts of the protocol.

The control flow perspective in this metamodel is represented by modeling elements such as `DeonticAssignment`, `Role` and `ActivityType`. The business partners in a business interaction play one or more roles of the business protocol. Deontic assignments stipulate under what conditions, each role has the permission, the obligation or the conditional obligation to perform certain activities. These activities are modeled using the `ActivityType` class. Only activity types that change the obligations, permissions and conditional commitments in a business interaction should be considered activities. In contrast, activities that merely correspond to obtaining data elements from another process participant, should not be considered activities. At implementation these issues can be hidden in the data layer. In Fig. 3 and Fig. 4 the metamodel of obligations and conditional commitments is represented in detail.

Deontic assignments are nearly always provided with deadlines on the performance of activities. Moreover, the performance of an activity always requires a performer and a beneficiary that receives the corresponding document. The information about deadline, performer and beneficiary, is captured in the concept of `Performance`. An obligation, permission or conditional commitment is initiated when an event performance occurs. The object of the obligation, permission or conditional commitment is represented by an obligation, permission or commitment performance. When a business partner makes a conditional com-

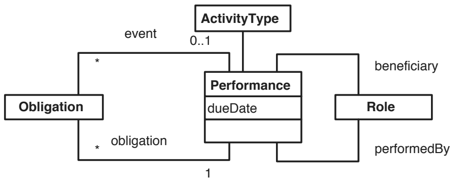

Fig. 3. A MOF/UML metamodel of an obligation

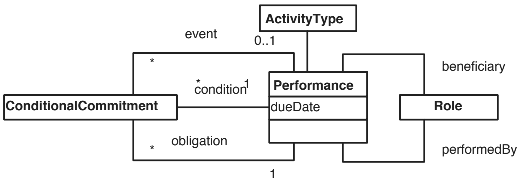

Fig. 4. A MOF/UML metamodel of a conditional commitment

mitment, he commits himself to the performance of an activity, on the condition of the performance of another activity. Similar to the concept of Performance, we have included the concept `Violation` into the metamodel, to represent the violations of obligations.

Example: Trade via a Trusted Intermediary. Companies might sometimes be reluctant to directly enter into a business transaction with an unknown trading partner. There is always a possibility that the opposite party will not observe its commitments: the buyer does not pay, the seller does not deliver within due time,... This risk is especially important in international trade, because of the high transaction costs involved. To resolve this stalemate companies usually work with one or more trusted intermediaries. These intermediaries, mostly financial institutions, act as a virtual serving-hatch between both parties. This modus operandi is applied with many different modalities, sometimes according to guidelines issued by international organizations such as the Practices for the Documentary Credit [22] of the International Chamber of Commerce.

For reasons of brevity, we base the running example of this paper on a business protocol for trade via a trusted intermediary which is described by Escrow.com [23]. The protocol proceeds as is informally depicted in Fig. 5. There are four roles in the protocol: a buyer, a seller, a trustee and a transporter. Initially, the buyer and seller negotiate the terms of the transaction. Secondly the buyer pays the trustee. Then the seller ships the merchandize, with the transporter as intermediary. Finally the trustee pays the seller, if the buyer accepts the merchandize. In Table 1 this business protocol is formally summarized using the conceptual model introduced in this section. Notice that the business protocol does not stipulate how the transporter gets paid, nor does it lay down which orders the seller is to accept and which to reject. This can be left to another

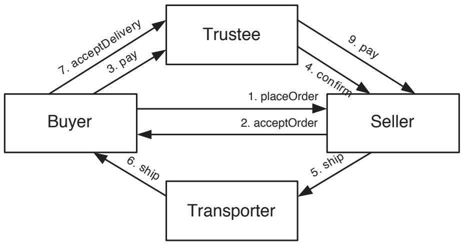

Fig. 5. An informal model of the Escrow.com business protocol

Table 1. Trade via a Trusted Intermediary

item	meaning
Role	buyer, seller, trustee, transporter
DocumentType	order document, proof of payment, shipment order,...
ActivityType	order, acceptOrder, rejectOrder, pay, confirmPayment, ship, makeDelivery, acceptDelivery, confirmDelivery, rejectDelivery
I1	An order must have at least one order line.
...	...
D1	The amount due to the seller is the total of the purchase order minus 1/8 percent commission.
...	...
DA1	Initially the buyer has the permission to place an order.
DA2	When the buyer places an order, the buyer is committed to pay the trustee, if the seller accepts the order.
DA3	When the buyer places an order, the seller must either accept or reject it within due date.
DA4	When the seller accepts the purchase order, the seller is committed to ship the goods within due date if the trustee confirms payment.
DA5	When the seller ships the goods, the transporter must make delivery.
DA6	When the customer pays the trustee, the trustee must confirm payment.
DA7	When the customer accepts/rejects delivery, the trustee must pay the seller/buyer.
DA8	When the buyer places an order, the buyer is committed to either accept delivery or reject the delivery with the trustee, if the transporter makes delivery.
DA9	When the buyer violates its obligation to either confirm acceptance or reject the delivery with the trustee within due date, the trustee must pay the seller.
DA10	When the seller violates its obligation to ship the goods with the transporter within due date, the trustee must pay the buyer.

business protocol and the business policy of the seller. Table 1 gives examples of the three kinds of business rules that are required to model the "trade via trusted intermediaries" business protocol. Rules numbered with **I**, **D** and **DA** are integrity constraints, derivation rules and deontic assignments respectively. From the metamodels the grammar of these rules should become apparent.

One of the main objectives of the business rules approach is to have business people write business rules [3]. In our opinion, this textual representation of Deontic Rules is comprehensible to end-users. Moreover, we know at least of one company, Unisys, that is working on a natural language parser to parse formal business rules written in natural language [24]. Decision Tables have been shown to be comprehensive representations of conditional derivation rules and reaction rules [4]. To date, however, no formal yet comprehensible graphical formalism exist for deontic assignments.

4 The Logical Background

4.1 Event Calculus

By performing or not performing the activities in the protocol, participants commit themselves to honor certain obligations in the future or grant the opposite parties in the contract the permission to perform other. The Event Calculus is an appealing logic to represent and reason about such deontic assignments.

The Event Calculus, introduced by Kowalski and Sergot [25], is a logic programming formalism to represent and reason about the effect of *events* and the state of the system expressed in terms of *fluents*. The Event Calculus is appealing for several reasons. For instance, the Event Calculus builds on a first-order predicate logic framework, for which efficient reasoning algorithms exist. In addition the Event Calculus has the ability to reason about time, in which fluents come to existence or cease to holds dynamically.

Shanahan provides suitable axiomatizations of the Event Calculus [26], in which the Frame Problem is solved through circumscription. Table 2 represents the predicates and axioms we have adopted to represent and reason about obligation and permission. In addition, we have operationalized the Event Calculus in CLP(fd), which allowed us to experiment with protocol verification and the transformation to decentralized business process descriptions.

4.2 Deontic Assignments in the Event Calculus

Deontic logic is a logic for representing and reasoning about deontic concepts such as obligation and permission. Various axiomatizations of deontic logic have been proposed. Føllesdal and Hilpinen present an approach built on propositional logic and call it Standard Deontic logic (SDL) [27]. SDL consists of an axiomatization using four operators for expressing obligations, permissions, prohibitions and waived obligations.

For the purpose of representing and reasoning about deontic assignments, SDL is a good starting point, but it lacks the following abilities. First of all, SDL

Table 2. Event Calculus predicates

term	meaning
$Initiates(\epsilon, \phi, \tau)$	event ϵ initiates fluent ϕ at time τ
$Terminates(\epsilon, \phi, \tau)$	event ϵ terminates fluent ϕ at time τ
$Initially_p(\phi)$	fluent ϕ holds from time 0
$Happens(\epsilon, \tau_1)$	event ϵ happens at time τ
$HoldsAt(\phi, \tau)$	fluent ϕ holds at time τ
$Clipped(\tau_1, \phi, \tau_2)$	fluent ϕ is terminated between times τ_1 and τ_2
$HoldsAt(\phi, \tau) \leftarrow Initially_p(\phi), \neg Clipped(0, \phi, \tau)$	
$HoldsAt(\phi, \tau_2) \leftarrow Initialles(\epsilon, \phi, \tau_1) \wedge \tau_1 < \tau_2 \wedge \neg Clipped(\tau_1, \phi, \tau_2)$	
$Clipped(\tau_1, \phi, \tau_2) \leftrightarrow \exists \epsilon, \tau[Happens(\epsilon, \tau) \wedge Terminates(\epsilon, \phi, \tau) \wedge \tau_1 < \tau < \tau_2]$	

is impersonal; it cannot express the partner to whom a particular obligation or permission applies. Another limitation is that SDL is static; it cannot represent deontic properties that come into effect and cease to hold because of timeouts or other events. Finally, it is desirable to have a logic that allows expressing so called contrary-to-duty obligations. These are reparative obligations that come into existence as the result of the violation of an obligation [28].

Governatori describes a defeasible deontic logic [29]. Several authors have built a Deontic Logic using the Event Calculus formalism [30] [16] [18] [31]. In these works deontic properties are represented as fluents, such that it is possible to represent and reason about the effects of activities on the obligations and permissions of actors. Table 3 enumerates the deontic fluents and axioms we use to represent Deontic Assignments.

Most authors consider the object of deontic properties to be activities, whereas Yolum and Singh use propositions [16]. This difference is at first sight negligible: some implementations of Deontic logic interpret deontic assignments as the obligation to bring about a certain proposition, others see it as the obligation to perform a certain activity. Another point of difference is the representation of timeouts on obligation or permission. Paschke and Bichler make use of continuous fluents to keep track of deadlines [31]. We provide all deontic properties with the δ argument. This representation is adapted from [18]. In axiom (2) we indicate that an obligation ceases to exist once the redeeming activity has been fulfilled within due time. A Violation of an obligation comes into existence if this obligation holds at due date, and the redeeming activity does not happen at due date. This is represented in (4). The representation and semantics of conditional commitments has been taken from [16]. Once the conditional activity happens, the conditional commitment becomes a base-level obligation, and the conditional commitment itself ceases to exist. This is represented in (5) and (6).

Notice that Deontic Logic usually also represents prohibition and waived obligations. For simplicity, we choose not to allow prohibition and waiver operators for modeling. Under a Closed-World Assumption these predicates can be made redundant in the following way. If no permission can be deduced, prohibi-

Table 3. Deontic properties

term	meaning
$Obligation(\pi, \alpha, \delta)$	partner π must do activity α by due date δ
$Permission(\pi, \alpha, \delta)$	partner π can do activity α prior to due date δ
$CC(\pi, \alpha_1, \delta_1, \alpha_2, \delta_2)$	partner π must do activity α_2 by due date δ_2 if activity α_1 is performed prior to due date δ_1
(1) $Permission(\alpha, \phi, \delta) \leftarrow Obligation(\alpha, \phi, \delta)$	
(2) $Terminates(\alpha, Obligation(\pi, \alpha, \delta), \tau) \leftarrow \tau \leq \delta$	
(3) $Terminates(\alpha, Permission(\pi, \alpha, \delta), \tau) \leftarrow \tau \leq \delta$	
(4) $Happens(Violation(Obligation(\pi, \alpha, \delta)), \delta) \leftarrow$	$HoldsAt((Obligation(\pi, \alpha, \delta)), \delta) \wedge \backsim Happens(\alpha, \delta)$
(5) $Initiates(\alpha_1, Obligation(\pi, \alpha_2, \delta_2), \tau) \leftarrow$	$\tau \leq \delta_1 \wedge HoldsAt(CC(\pi, \alpha_1, \delta_1, \alpha_2, \delta_2)), \tau)$
(6) $Terminates(\alpha_1, CC(\pi, \alpha_1, \delta_1, \alpha_2, \delta_2), \tau) \leftarrow$	$\tau \leq \delta_1$

tion is explicitly assumed. If no obligation can be deduced, waiver is explicitly assumed. To conclude, we give in Table 4 the transcription of some of the deontic assignments of the example introduced in section 3.

Table 4. Trade via a Trusted Intermediary

DA1	$Initially_p(Permission(Buyer, PlaceOrder(Buyer, Seller))$
DA2	$Initiates(PlaceOrder(Buyer, Seller),$ $CC(Buyer, AcceptOrder(Seller, Buyer), \tau + 2,$ $Pay(Buyer, Trustee), \tau + 4), \tau)$
...	
DA10	$Initiates(Violation(Obligation(Seller,$ $Ship(seller, transporter), \delta_1)),$ $Obligation(Trustee, Pay(Trustee, Buyer), \tau + 1), \tau)$

5 Verification of Business Protocols

The introduction or modification of a business protocol can have far reaching consequences for all members of the business community. Because business rule-based business protocols are the starting point for the design of a business' private business processes, it is of paramount importance that the formal business protocol specification contains no errors. Fortunately, the rich semantics of the commitment-based business protocols and the availability of efficient reasoning procedures present new opportunities for verification and validation. Without going into detail, we can highlight the following verification issues:

- **Rule conflicts**: Rule conflicts exist when sets of rules, such as conditional integrity constraints or derivation rules do not consider all possible cases or when rules in a rule set contradict or subsume each other. To detect and avoid such problems, efficient anomaly checking methods have been developed [32].
- **Deadlocks and livelocks**: In a deadlocks situation, no permissible performance can carry the business interaction forward such that a new state of permissions, obligations and conditional commitments exist. Such a scenario might consist of two business partners having conditional commitments towards each other, but the conditional performance to turn at lease one of these conditions into a base-level obligation is not permitted. For example, the buyer has made the conditional commitment to pay upon delivery, whereas the seller has made the conditional commitment to deliver upon payment. In a livelock situation, the protocol state is trapped in an infinite loop. Notice that it is not the occurrence of a loop that defines the livelock, but the occurrence of loops without a permissible performance that leads to a deontic state outside the loop. Algorithms should be developed to ensure that each legal scenario leads to termination, a state in which no obligations or permissions exist.
- **Deontic conflicts**: Deontic conflicts arise when there are protocol states in which a business partner has both the permission and the prohibition to a performance or when he has both an obligation and obligation waiver to a performance. Such deontic conflicts can be detected through abductive reasoning in the Event Calculus. We have experimented with ASystem, a system for abductive reasoning within the framework of Abductive Constraint Logic Programming [33]. The issue here is to ask the reasoner if it can come up with a narrative of performances such that a protocol state arises in which such a deontic conflict exist. Note, however, that in the grounding of deontic assignments in this paper it is not possible to have deontic conflicts, because we do not allow to use prohibition and waiver modeling constructs.
- **Temporal conflicts**: Temporal conflicts occur when two deontic assignments at the same time initiate and terminate a permission, obligation or conditional commitment. This anomaly can easily be detected by querying the set of assignment rules.
- **Trust conflicts**: In a business interaction trust is an important issue. When a business wants to incorporate a new business protocol into it's own business processes, it must be sure that newly possible business interactions put the business in a position were it has direct obligations towards non-trusted business partners that involve sensitive activities such as payment or the shipment of goods, that are not neutralized by preceding performances of the opposite party.

6 Generating Compliant Business Process Descriptions

Companies can use the formal expressions of business protocols to generate compliant business processes or to verify existing business process models for compliance. By definition, a business protocol only prescribes the public aspect of a business interaction. As a consequence, business partners have, within the regulatory framework of a business protocol, the autonomy to perform activities according to their own business policies. We have constructed an algorithm, called PENELOPE, (Process Entailment from the Elicitation of Obligations and Permissions) that generates these process flows for each business partner, by applying deductive reasoning over the Event Calculus. In this section we indicate the transformation principles behind this algorithm.

Figure 6 represents the public aspect of the process models of the `buyer`, `seller`, `trustee` and `transporter` roles of the example introduced in section 3. The notation used is based on the rule-based process models we discuss in [4]. Although some visual elements correspond to other process notations, our notation is different, mainly because it distinguishes decision points in a process description, which are represented as circles in the diagram. *Decision points* group sets of reaction, or Event-Condition-Action rules. The events in such rules are the inflowing arrows into a decision point. The activity, that can be launched in response to a certain combination of business events are represented with the arrows that flow out of a decision point into the activity type rectangles. To distinguish each business partner's process model, we have adopted the swim lane model element.

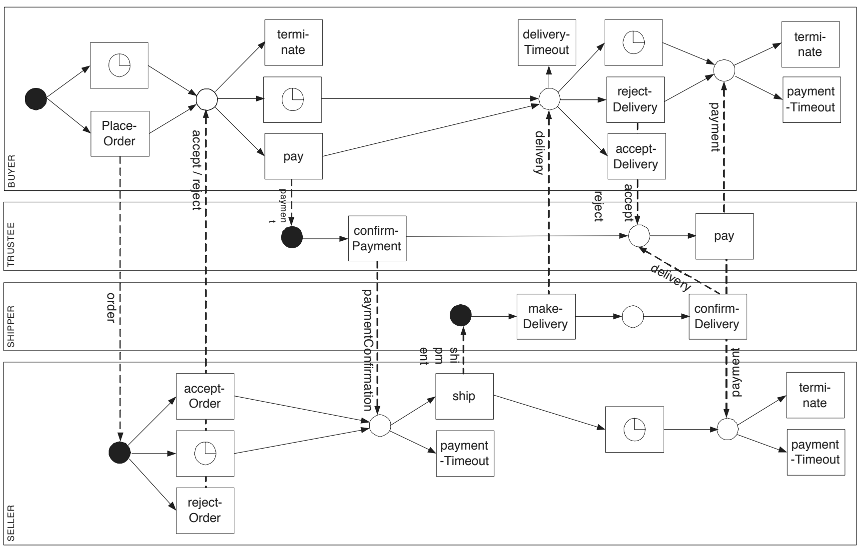

Fig. 6. Trade via a Trusted Intermediary: four process models

As mentioned previously the "trade via trusted intermediaries" business protocol does not specify under what conditions the seller is to accept or reject a purchase order. Indeed, it seems most appropriate to leave this freedom of choice to the business policy of the seller. For example, the seller might have the business policy to accept each purchase order, provided that the ordered goods are available. This aspect of business logic can be modeled using reaction rules graphically displayed in the decision table in Fig. 7 [4]. Notice that in the transformation of business protocols to business process models, the reception of business documents could correspond to external business events, whereas the expedition of business documents correspond to the notification of a performance.

reaction rule			
event: PlaceOrderEvent(po)	Y		N
condition: Available(po)	Y	N	-
action: AcceptOrder(po)	x	-	-
action: RejectOrder(po)	-	x	-

Fig. 7. Reaction rule: acceptation policy

A process model must not violate the obligations for which no violation is allowed. But, by way of precaution, it must foresee the possibility that other business partners violate obligations. For instance, if a buyer places an order, it must foresee never to receive a rejection or acceptance from the seller. Violations of obligations can only be detected if the due dates on obligations are tracked during process execution. This is represented in the process models using the timeout activity types, which are displayed as timer rectangles. Sometimes a protocol might specify a reparative obligation, a contrary-to-duty obligation, which resolves the violation. If, however, a contract violation for which no reparation exists is detected, the process instance is most likely to be terminated. In such cases, the deontic conflicts between business partners might be resolved through human interaction.

7 Conclusion

In this paper we set out to make the knowledge, implicitly present in business protocols and business policies, explicit as business rules to enhance the flexibility of Information Systems. To this end, we have outlined a lightweight business protocol metamodel that can be used to capture the semantics of arbitrary business protocols. To formalize the semantics we have grounded deontic assignments in formal logic, relying on concepts and formalisms that have been developed by the Logic Programming community. In addition we have highlighted the relevant issues in the verification of business protocols and have given indications how compliant business processes can be generated from business protocol specifications.

With the word business rule existing for over a decade, and the concept existing for presumably much longer, it is in our opinion deplorable that so little of the principles of business rules have been realized in commercial software. Admittedly, there are still many important problems to be solved. Nonetheless we believe that fundamental research in domains as web services, logic, ontologies and multi-agent systems can provide a fruitful breeding ground for evolutions in business rule modeling and will eventually lead to comprehensive service-based platforms that demonstrate their full potential.

References

1. Wagner, G.: The agent-object-relationship metamodel: towards a unified view of state and behavior. Inf. Syst. **28**(5) (2003) 475–504
2. Herbst, H.: Business Rule-Oriented Conceptual Modeling. Physica-Verlag (1997)
3. Ross, R.G.: Business Rule Concepts - Getting to the Point of Knowledge. Business Rules Solutions, LLC; 2nd edition (2005)
4. Goedertier, S., Vanthienen, J.: Rule-based business process modeling and execution. In: Proceedings of the IEEE EDOC Workshop on Vocabularies Ontologies and Rules for The Enterprise (VORTE 2005), Enschede (2005)
5. Morgan, T.: Business Rules and Information Systems: Aligning IT with Business Goals. Addison-Wesley Professional; 1st edition (2002)
6. Kardasis, P., Loucopoulos, P.: Managing business rules during the requirements engineering process in rule-intensive IT projects. In: Proceeding of the 6th International Conference on Business Information Systems (BIS 2003), New York, NY, USA, Springer (2003)
7. Wan-Kadir, W.M.N., Loucopoulos, P.: Relating evolving business rules to software design. Journal of Systems Architecture **50**(7) (2004) 367–382
8. OMG: Semantics of Business Vocabulary and Business Rules (SBVR) (2005) Available to OMG members as bei/2005-08-01 at `http://www.omg.org`.
9. Bussler, C.: The role of b2b protocols in inter-enterprise process execution. In: TES '01: Proceedings of the Second International Workshop on Technologies for E-Services, London, UK, Springer-Verlag (2001) 16–29
10. Peltz, C.: Web services orchestration and choreography. IEEE Computer **28**(10) (2003) 46–52
11. Singh, M.P., Chopra, A.K., Desai, N., Mallya, A.U.: Protocols for processes: programming in the large for open systems. SIGPLAN Not. **39**(12) (2004) 73–83
12. Yolum, P., Singh, M.P.: Commitment machines. In Meyer, J.J.C., Tambe, M., eds.: ATAL. Volume 2333 of Lecture Notes in Computer Science., Springer (2001) 235–247
13. Yolum, P.: Towards design tools for protocol development. In: AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, New York, NY, USA, ACM Press (2005) 99–105
14. Kabilan, V., Zdravkovic, J., Johannesson, P.: Using multi-tier contract ontology to deduce contract workflow model for enterprise interoperability. In Missikoff, M., Nicola, A.D., eds.: EMOI-INTEROP. Volume 160 of CEUR Workshop Proceedings., CEUR-WS.org (2005)
15. Grosof, B.N., Poon, T.C.: Sweetdeal: representing agent contracts with exceptions using xml rules, ontologies, and process descriptions. In: WWW '03: Proceedings of the twelfth international conference on World Wide Web, ACM Press (2003) 340–349

16. Yolum, P., Singh, M.P.: Reasoning about commitments in the event calculus: An approach for specifying and executing protocols. Annals of Mathematics and Artificial Intelligence **42**(1-3) (2004) 227–253
17. Rouached, M., Perrin, O., Godart, C.: A contract layered architecture for regulating cross-organisational business processes. In van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F., eds.: Business Process Management. Volume 3649. (2005) 410–415
18. Knottenbelt, J., Clark, K.: An architecture for contract-based communicating agents. In: Proceedings of the Second European Workshop on Multi-Agent Systems. (2004)
19. Fensel, D., Bussler, C.: The web service modeling framework wsmf. Electronic Commerce Research and Applications **1**(2) (2002) 113–137
20. Orriëns, B., Yang, J., Papazoglou, M.P.: A rule driven approach for developing adaptive service oriented business collaboration. In Benatallah, B., Casati, F., Traverso, P., eds.: ICSOC. Volume 3826 of Lecture Notes in Computer Science., Springer (2005) 61–72
21. Halpin, T.A.: A Fact-Oriented Approach to Business Rules. In: ER. (2000) 582–583
22. ICC: UCC uniform customs and practice for documentary credits and Supplement to UCP 500 for electronic presentation (eUCP) version 1.0. (2002)
23. Escrow.com: Guiding buyers and sellers safely through the online escrow process. `www.escrow.com/solutions/escrow/process.asp` (2006)
24. Baisley, D.: OMG and Business Rules. Presentation available at `http://www.omg.org/docs/omg/05-04-09.pdf` (2005)
25. Kowalski, R., Sergot, M.: A logic-based calculus of events. New Gen. Comput. **4**(1) (1986) 67–95
26. Shanahan, M.: Solving the frame problem: a mathematical investigation of the common sense law of inertia. MIT Press, Cambridge, MA, USA (1997)
27. Føllesdal, D., Hilpinen, R.: Deontic logic: An introduction. In Hilpinen, R., ed.: Deontic Logic: Introductory and Systematic Readings. D. Reidel Publishing Company, Dordrecht (1971) 1–35
28. Governatori, G., Rotolo, A.: A gentzen system for reasoning with contrary-to-duty obligations. a preliminary study. In Jones, A.J., Horty, J., eds.: Δeon'02, London, Imperial College (2002) 97–116
29. Governatori, G.: Representing business contracts in *RuleML*. Int. J. Cooperative Inf. Syst. **14**(2-3) (2005) 181–216
30. Marín, R.H., Sartor, G.: Time and norms: a formalisation in the event-calculus. In: ICAIL '99: Proceedings of the 7th international conference on Artificial intelligence and law, New York, NY, USA, ACM Press (1999) 90–99
31. Paschke, A., Bichler, M.: SLA Representation, Management and Enforcement. In: EEE '05: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05), Washington, DC, USA, IEEE Computer Society (2005) 158–163
32. Mues, C., Vanthienen, J.: Efficient rule base verification using binary decision diagrams. In Galindo, F., Takizawa, M., Traunmüller, R., eds.: DEXA. Volume 3180 of Lecture Notes in Computer Science., Springer (2004) 445–454
33. Van Nuffelen, B., Kakas, A.C.: A-system: Declarative programming with abduction. In Eiter, T., Faber, W., Truszczynski, M., eds.: LPNMR. Volume 2173 of Lecture Notes in Computer Science., Springer (2001) 393–396