University of Thessaly

The Representation of Electronic Agreements: a Survey of Logic-Based Approaches

Georgios K. Giannikis and Aspassia Daskalopulu

Department of Computer and Communications Engineering,
University of Thessaly,
38221 Volos, Greece
{ggiannik,aspassia}@inf.uth.gr

Abstract. The term "e-contract" is used to refer to a legally binding agreement that is settled between two or more parties in electronic marketplaces. This report surveys the main logic-based approaches to the representation of e-contracts that have emerged during the last decade; most have been proposed as appropriate solutions for contract performance monitoring. We broadly distinguish three application areas, where electronic agreements play a central role: e-commerce, business process modeling and automation, and virtual communities. First we note a set of requirements that a representation of electronic agreements should meet, in order to facilitate the development of tools for contract performance monitoring. We then present the characteristic techniques that have been employed by the various approaches and comment on the extent to which they meet these requirements. Finally, we present separately our approach and comment on the issues that it seeks to address.

Keywords. E-contracts, E-marketplaces, Contract Representation, Contract Performance Monitoring

March 2007

Table of Contents

1 Introduction

During the last decade a rapid growth on research activity related to e-marketplaces has occurred. Such domains were considered as open electronic environments that provide a framework for brokering, negotiation, contract establishment and subsequently agreement monitoring. The term "e-contract" is used to refer to a legally binding agreement between two or more parties. Generally, an e-contract creates mutual legal relations between the parties involved and determines what actions are obligatory/permissible/forbidden to be performed by the parties.

The whole contractual activity can be regarded of comprising of two different phases, namely *contract formation* and *contract performance*. During the former, parties communicate with each other, exchange needed business information and negotiate the terms and conditions of their agreement. During the latter, once an agreement has been established; contract execution takes place, during which violations of contract clauses may occur, thus raising the need for reparatory mechanisms to be deployed, if any such are stipulated in the agreement.

In this report we review research efforts related to contract performance monitoring that have emerged during the last decade and are based on logic. Typically each approach involves a representation of deontic notions (such as obligation, permission, prohibition and power), their associated meta-level notions (such as violation, sanction, compliance and normative conflict), mental notions (such as beliefs, desires, intentions and trust) and domain-independent concepts such as time, actions and their effects.

The rest of the report is organized as follows: in section 2 we present an example scenario and record requirements for efficient reasoning with electronic contracts; in section 3 we survey all gathered approaches from the literature during the last decade; in section 4 we present our approach to e-contracting and discuss its relation to other approaches; and finally, in section 5 we present our conclusions.

2 Requirements for e-Contract Performance Monitoring

For the purposes of illustration consider a 3-party business transaction that takes place in an electronic marketplace populated by software agents. A buyer agent (BA) communicates with a seller agent (SA) and establishes an agreement for purchasing a certain product. Consequently, SA communicates with a carrier agent (CA) and establishes another agreement for the timely and safe delivery of goods to BA.

The first agreement (between BA and SA) is to be conducted on the following terms: SA should see to it that the goods be delivered to the BA within 10 days from the date BA's order happens. BA, in turn, should see to it that payment be made within 21 days from the date it receives the goods. If SA does not deliver on time, then a fixed amount is to be deducted from the original price of the goods for each day of delay and it should see to it that delivery be made by a new deadline, say within the next 3 days. If BA does not perform payment on time, then a fixed amount is to be added to the original price of the goods for each day of delay and it should see to it that payment be made by a new deadline, say within the next 5 days. In the same spirit, the second agreement (between SA and CA) defines obligations, deadlines and possible sanctions/reparations in case of violations.

Following [13], we may take an informal, process view of the business transaction that is regulated by the two agreements. Each state offers a (possibly partial) description of the factual and normative propositions that hold true in it. A transition between states corresponds to an event that takes place, i.e., an action that one of the parties performs or omits to perform. Initially, at time point T0, the transaction is in state S0 where the two agreements have been established and no events have occurred yet. If BA places an order at some time after T0, the transaction will move to a state S1, where SA is obliged towards BA to deliver goods within 10 days. Also, CA's obligation towards SA, to deliver goods to BA on SA's behalf within 10 days, is active. If CA delivers within the specified time bounds, then the business exchange will move to a state S2, where CA's obligation (and SA's obligation towards the BA for delivery, which is related to it) is successfully discharged, and BA's obligation towards SA to pay becomes active (as does SA's obligation to pay CA). If, when the transaction is at state S1, CA does not deliver on time, then the transaction will move to some state S3, where SA must compensate BA as specified by their agreement (and CA must compensate SA as specified by their agreement). In the same manner we may discuss other states of the business exchange.

Although this scenario is superficially simple, it puts on view several important features that are met on more sophisticated scenarios and contractual business transactions, and motivates a discussion about the requirements for e-contract representations. Such requirements include, for instance:

1. Ontology representation: The concepts of the application domain as well as their relations need to be represented explicitly, so that such information may be used during inference.

2. Temporal information representation: We need a proper formalization of time and temporal information, as noted by many researchers, for instance [82]. Here are some examples that show that temporal reasoning is required. Expressions such as:
"BA places an order at time point T", or
"SA is obliged towards BA to deliver ordered goods within 10 days from the date BA's order happens", or
"At state S1 the CA agent is obliged to delivery goods to the BA agent by time point T+10"
show that we are not only interested in the actual time points at which an action happens or a (normative or descriptive) proposition holds but also in deadlines. Indeed, specifically for norms [51] note the distinction between their so called *internal* and *external* times. We may also need to represent periodic information, for instance:
"RA is obliged to perform installment payment each month".
Specific issues that arise concern the commonsense law of inertia, the representation of the indirect effects of actions, the representation of non-deterministic actions, concurrent actions and so on.

3. Deontic Modalities: We need to determine what normative relations obtain between parties during a business transaction. Deontic Logic studies such notions, i.e., obligations, permissions, prohibitions and their interrelation [11, 55]. Deontic Logic allows us to disconnect what is the case from what ought to be the case. Such distinction enables us to determine explicitly whether the actual behavior of contractual parties complies with the prescribed behavior. [43, 84, 56].

4. Legal and Physical ability: We need to distinguish between the legal/institutional and the actual/physical ability of involved parties to perform actions in order to meet their obligations [50, 44]. Such notions are essential, because they affect which actions are considered as valid and consequently which actions' effects obtain the domain [44].

5. The representation of normative violation: In realistic domains, such as electronic marketplaces, along with the notion of obligation comes the notion of violation. Their relation is obvious. Given a specifications of the agents' obligations during a business transaction, which typically involve deadlines, the e-contract representation ought to facilitate the automated determination of agreement violations. There are many ways in which an agent may violate its obligation to perform an action A by time T, e.g. the agent may perform A but not within the deadline, or the agent may not perform A at all, or the agent may perform some other action B which renders performance of A impossible etc. Similar concerns arise in the case of prohibitions, which, again, may be violated in more than one ways.

6. Contrary to Duty Structures: Contrary to duty structures are the specification of a primary obligation, along with the specification of a secondary obligation that obtains if the primary one is violated. One may see them as a priori (to contract violation) determinations of recovery mechanisms [12, 61].

7. Normative conflict representation: According to [75] a conflict arises when "(possibly) valid norms establish incompatible qualifications for the same concrete state". A norm set may be either inconsistent, if a contradiction is logically derivable from it, or potentially inconsistent, if it may lead to contradiction in an upcoming state. In similar spirit in [40, 41] moral conflicts are defined as states where an agent ought to do an action A and, at the same time, it ought to do another action B, but it is impossible to do both. Such situations are often met in business transactions where agents either are in conflict and need a resolution or face a potential conflict and need a plan to overcome this situation or to deal with it in a self-serving manner.

8. Default, Defeasible and Nonmonotonic Reasoning. The need for reasoning by default, defeasibly and nonmonotonically in legal domains is strongly argued in many research papers, i.e., [76, 18, 10, 37, 60] among others. Different dimensions and interpretations of this kind of reasoning have been discussed in various approaches with respect to the underlying logic that each approach adopts.

9. Auxiliary calculations. We need to be able to define and use formulae and procedures that enable the dynamic calculation and re-recalculation of domain concepts, such as deadlines based on relative times, or amount of money for payment.

3 Main Techniques for e-Contracting

In this section we present research efforts that are based on logic and have been proposed during the last decade. The research community that has been concerned with electronic agreements has focused not only on e-commerce applications but also on business process modeling and automation and social norms that govern virtual communities. Our aim is to commit to this report the fundamental features that each proposed framework contributes. We classify all gathered research approaches in subsections based on the application domain (e-commerce, e-business, virtual communities) and the specific characteristics/features that distinguish them from other frameworks. In subsections 3.1 to 3.5 we record approaches that refer to e-commerce applications, while in subsection 3.6 we record approaches that refer to applications for business process modeling and automation and virtual communities.

3.1 Dynamic Logic

In this part we discuss efforts presented in [72, 13, 14, 71, 70]. Those approaches are grouped together due to the common inspiration, which originate from Meyer's dynamic logic formalisation of deontic notions [54], to address reasoning with time and actions.

Santos and Carmo, in [72], propose a set of deontic operators in order to specify the intended set of behaviors that are related with contractual parties. Deontic operators are combined with dynamic operators to represent actions. Furthermore operators present a temporal dimension through their semantics. Desirable behaviors of contractual parties are based on the concept of obligation. Obligations were defined by a special kind of norm, in the juridical context, called prescription. According to [83] a prescription is a command or permission, settled by someone in an authoritative position, towards agents with the intension of inducing or allowing them certain behaviors or conducts. Obligations are examined from the point view of their fulfillment and/or their violation through an a posteriori verification of the actual behavior. On the whole, the proposed logic mainly addresses compliance with the agreement that is requirements 2, 3 and 5. This work is the first approach in the analysis and representation of contractual obligations and set the basis for subsequent proposals.

In [13, 14] a Modal Action Logic combined with Deontic Logic operators approach was proposed by Daskalopulu *et al.* A contract is modeled as a process whose state at a given time is determined by the legal relations that stand between contractual parties. Transitions between states are affected by parties' actions. Depending on whether parties' actions comply or violate contractual behaviors, the resulting state is defined as acceptable or unacceptable. An unacceptable state either sets the abnormal ending of the business transaction or is unacceptable in a tolerable way because reparation is possible. Moreover, reparations associated with the violation of obligations are studied and a suitable representation of contrary to duty structures in fault tolerant systems is proposed. To sum up, this approach mainly concerns requirements 2, 3, 5 and 6.

In [70, 71, 69] an electronic agent-based contract framework layered on top of existing B2B frameworks is presented by Sallé *et al.* The framework was designed to support the whole life-cycle of the contract, which consists of three phases: (i) drafting, (ii) formation, and (iii) fulfillment. In this work, contracts were defined as sets of statements of participant's intentions. The contract specifies the behavior of contractual parties in ideal worlds as well as in sub-ideal worlds where parties' do not fulfill their commitments. Contract structure was separated in two main parts: (i) an informative section that contains information such as identification number, identities and roles, validity period and a normative system of reference, and (ii) a behavioral specification section which is a set of informative statements that describes the expected behavior of participants. All contractual obligations are associated with sanctions. This characteristic gives the agent the advantage of a deliberative decision on fulfilling or not a normative statement based on positive or negative effects. Two types of sanction norms were proposed. Endogenous sanctions, i.e., contrary to duty structures, and exogenous sanctions that apply when violations with no specific endogenous sanction occur. To sum up, this approach mainly concerns requirements 2, 3, 5 and 6.

3.2 Event Calculus

Here we discuss two frameworks presented in [23, 46]. Both approaches adopt a contract representation in Event Calculus (EC) for temporal reasoning and reasoning with actions and their effects [47, 77]. EC is also used in [90, 66, 67] but we discuss these approaches later in subsection 3.4 because they have another important distinguishing feature.

In [23] Farrell *et al.* present an ontology and a tool to capture issues that are related with contract state tracking for Service Level Agreements. The presented framework is implemented using the Java programming language and is constructed on an XML-based formalization of the Event Calculus, called ecXML. Their main intention is the implementation of a tool, called Event Calculus State Tracking Architecture (ECSTA), which is able to track the effects of various events on different contractual states and to define what normative relations hold between parties on those states. Moreover, a detailed discussion about notions such as obligation, permission (vested permission) and institutionalized power (vested power) and their role in the business transaction is presented. Based on the above analysis, three types of contractual norms are proposed: contract management norms, obligation norms and privilege norms. We discuss the third type which concerns with actions that are permitted to be performed and are not explicitly recorded in the contract. According to Farrell *et al.* any action that is not permitted is considered to be an illegal action. This fact leads us to the conclusion that there is no need for explicit prohibition norms or in other words the absence of permission is considered as the presence of prohibition. On the whole, requirements 2, 3, 4 and 9 are explicitly discussed; we believe that requirements 5 and 6 are also met by this approach, although the authors do not explicitly discuss them.

Knottenbelt and Clark, in [46], introduce a simple Event Calculus representation of contracts and a Belief-Desire-Intension (BDI) architecture that supports contract performance. The proposed architecture was built on top of the AgentSpeak(L) [62] agent architecture and enables agents to respond to events in a reactive or a proactive manner based on their active contracts and temporal conditions. During this work two types of contracts were studied. Short-term contracts like the one presented in our example scenario and long-term contracts that define requirements of short-term contract drafting. Communication between agents is possible by exchanging events. An event is considered as the act of sending messages. A well-formed message consists of a time stamp, an identifier, the identifier of the message to which it is a reply, a sender, a receiver, content, context, and the interaction protocol. During messaging exchange an agent is able to evaluate a contract by placing a query on the Event Calculus HoldsAt predicate [47, 77]. To sum up, this approach deals with requirements 2, 3 (only obligations are discussed), 4 (only institutionalized power is discussed) and 5. Moreover, the authors claim that conflict detection and resolution is possible through the work presented in [8].

3.3 Nonmonotonic Reasoning

In this section we discuss approaches that deal with defeasible and nonmonotonic reasoning with e-contracts [35, 63, 36, 34, 4, 29, 30, 59, 58]. In what follows we present the main points of those proposals. Note that although the underlying logic language and theory are different, these approaches present many common features due to the interrelation of the adopted logics.

In a series of papers, such as [35, 63, 36, 34, 4] Grosof *et al.* presented a comprehensive approach to the representation of business rules and a series of tools that are integrated in the WWW framework. Specifically, in [35] a declarative approach to the representation of e-contracts rules that is based on Courteous Logic Programs (CLP) is introduced. CLP is an extension of Ordinary Logic Programs (OLP) with prioritized conflict handling. The central purpose of this work is to present declarative contract semantics, to handle potential conflicts with priorities, to represent contract rules with an XML-based encoding and to present a prototype called Common Rules. This works is mainly concerned with the contract negotiation phase and particularly with a suitable contract rule representation for communication during this phase. Moreover, in [35] an XML formalism of CLP rules called Business Rules Markup Language (BRML) and a prototype implementation named Common Rules was also introduced. This work was extended in [63] and an auction-based negotiation tool called ContractBot was introduced. Here a contract representation in CLP rules that consists of two subsets is presented. The first subset, called, proto-contract, contains rules that determine facts and conditions of the overall transaction, such as ways of delivery, payment or reparation, while the other subset contains negotiation-level rules, that describes of what and how will be negotiated. In [34] an overview of all previous efforts is available plus an extension, of the previous work on business rules representation, which is based on Situated Courteous Logic Programs (SCLP) is also introduced. SCLP is the Situated extension of CLP that is characterized of features such as nonmonotinicity, that are negation as failure and prioritized conflict handling as presented above and furthermore, procedures for querying on contracts and representing actions. Note that conflict detection is facilitated with the use of mutual exclusions statements, which are statements (pair of literals) that determine contradictory or inconsistent transaction states. On the whole, this approach deals well with requirements 5, 7, 8 and 9, but no temporal representation was adopted in order to facilitate reasoning with time.

In [29, 30] an architecture to represent and reason about e-contracts is introduced by Governatori *et al.*. The system is called DR-Contract and extends the DR-Device architecture (a system for defeasible reasoning on the Semantic Web [3],) with the Defeasible Deontic Logic of Violation (DDLV) [29, 32]. The aim of this approach is to analyze the expected behavior of the contractual parties and to identify what normative relations arise from an e-contract. Contracts are considered to comprise provisions that determine obligations, permissions, entitlements and other mutual normative positions that hold among contractual parties. Contract clauses are separated in two different types: (i) definitional clauses that define contractual concepts such as "who is a privileged customer" or "what is a special order", and (ii) normative clauses that contain deontic notions and intend to regulate the whole transaction. The underlying logic that is adopted is Nute's Defeasible Logic [57]. According to this theory four types of knowledge are considered: (i) facts, (ii) strict rules, which are rules in the classical sense (iii) defeasible rules, which are rules that can be defeated by other rules, and, finally, (iv) superiority relations, which define priority relations among rules. Another point worth mentioning is the fact that this approach also deals, in detail, with the issue of violation of primary obligations and their reparation mechanisms. Contrary to duty structures were represented by introducing a new non-classical connective $\otimes$ [29, 32]. The interpretation of the formula $OA \otimes OB$ is "Obligation B is the reparation of the violation of obligation B". This connective allows the combination of primary and reparatory obligation in a single regulation and satisfies important properties such as associativity, duplication and contraction on the right that enable reasoning with CTDs. Note that, according to Governatori, the Courteous Logic Programs of the previous presented approach is a notational variable of Defeasible Logic and thus the integration of properties of both approaches is possible. To sum up, this approach covers requirements 3, 5, 6, 7 and 8. No temporal dimension is given via the integration of some temporal logic, but an extension to this direction is feasible as shown in [31].

In [59, 58] Paschke *et al.* presented the ContractLog system. This approach deals with execution and monitoring of Service Level Agreements. SLAs are represented via Event-Condition-Action rules that are enhanced with EC predicates and other special predicates for deontic notions. This work, also, adopts Nute's Defeasible Logic and presents ideas that are similar to both previous presented approaches. As very little technical detail is available, this approach seems to address requirements 2, 3, 5, 6, 7, 8 and 9, but we have not been able to verify this.

3.4 Commitments

In this section we have gathered and discuss approaches that see e-contracts from a commitment-based perspective [86, 87, 88, 89, 90, 66, 67, 48]. Although the perspective is similar, they vary in what commitments denote.

In [86, 87] monitoring requirements for e-marketplaces and a system architecture are presented. Specifically, in [87], Xu proposed an approach for contract modelling that is based in Temporal Logic. This work aims to facilitate pro-active monitoring and violation prevention. This is accomplished by proposing workflow constrains and guards of workflow constrains that describe different complex relationships among actions and make possible to take the initiative to anticipate and avoid contract violations. Moreover a guard and a pro-active detection algorithm are presented to dynamically monitor business processes. Supplementary to previous papers, in [88] the notion of commitments is added to the formal representation of the electronic contract. A commitment is considered not as a distinct obligation but as a guarantee by one party towards other parties that some action sequence shall be executed completely. This fact is the main difference with the next three approaches. Next to the notion of commitment, the commitment graph is presented that is an overview of commitments between agents. So the commitment graph is a graphical encoding of contract clauses. This graph in cooperation with the two algorithms may point out which partner is responsible for which violations as shown in detail in [89]. On the whole, this approach deals with requirements 2, 3 (via commitments, not via classical deontic notions) and 6.

Yolum and Singh presented in [90] an approach for specifying and executing protocols that regulate multi agent interactions. Such protocols define a set of social commitments (or else commitments) that are assigned to agents. Conceptually, commitments capture obligations arising for an agent towards another agent to bring about a certain property. The business transaction is viewed as a finite state machine where operations (actions) on commitments and business rules are being represented in the circumscriptive version of the Event Calculus language as explained in [78]. Two basic commitment types are considered [79]: (i) Base-level commitments meaning that an agent is committed towards another agent to bring about condition, and (ii) Conditional commitments meaning that if a condition is satisfied then an agent will be committed towards another agent to bring about another condition. Different operations on commitments are defined which we do not discuss here due to space restrictions. Possible transitions in the business protocol can be specified in terms of the Event Calculus language. Once again this approach addresses requirements 2, 3 and 5 and moreover it addresses nonmonotonic reasoning through the circumscriptive version of the Event Calculus.

In [66] and [67] Rouached *et al.* present (i) a layered contract model, (ii) an approach for regulating Web Services to support cross-organizational collaborations, and (iii) how the integration of contact management services into the overall business process may be facilitated. This work, also, uses the Event Calculus language as presented in [47] to specify the contract state at particular time points. A point that is worth mentioning is that, special terms, expressing temporal relations, are used to express the relation between the occurrences of different events (composite events).

As in [79, 90] this work accepts three types of commitments. The third type is the Persistent commitment expressing that an agent is committed towards another agent that some condition holds on all future time points. Here, deontic clauses, such as obligation, permission and prohibition, are defined in terms of operations on commitments using both commitments and EC axioms [78]. With respect to the specified requirements, this approach deals with issues number 2 and 3.

In [48] another approach that considers contracts as protocols that regulate business agreements by specifying a set of commitments is proposed by Letia and Groza. Contracts are represented by Defeasible Commitments Machines (DCM), which is a theory in the Normative Defeasible Logic (NDL) presented in [31]. The theory consists of two parts. The first part captures the representation of standard commitments and the possible operations on them in terms of (NDL). The second part includes all contract dependent rules. As in previous commitment-based approaches, this work accepts two types of commitments (Base-level and Conditional commitments). Temporalized Defeasible Logic in combination with time constraints for commitments (deadlines for fulfillment) facilitates the entailment of conclusions about commitment states over time. In this way, besides the gain of reasoning temporally, agents are also able to reason with incomplete knowledge. To conclude, this approach addresses requirements 2, 3 but there is no mention about permission or prohibition, 7 but there are no particular conflict patterns specified, and 8 via Defeasible Logic.

3.5 Linguistic Aspects of e-Contracts

In this section we discuss the work presented in [80, 81]. The particular feature of the work of Tan and Thoen is the fact that it specifies the need for directed deontic notions. It deals with e-contracts from a linguistic perspective and, therefore, this approach mainly concerns with issues 1[1] and 3. [80] addresses some unanswered questions of their previous work where a formal model, called Deontic Deep Structure Model, was presented. According to [80] (i) the ambiguities that derive from the underlying logic for directed obligation [38, 73], as adopted in their previous work, and (ii) its shortcoming to express directed permissions, raised the need for improvement. An alternative definition for directed obligations is presented and a definition for directed permission is proposed. These definitions are based on a conditional operator interpreted as "count as" and an attempt operator, as presented in [44, 74], respectively. Moreover, a different interrelation between directed obligation and directed permission form the one that holds in Standard Deontic Logic for obligation and permission is proposed. In their later work, [81], an approach to deal with requirement 2 is presented. Towards this direction a contract representation in the Formal Language for Business Communication (FLBC) [45] was proposed.

[1] Note that without any distinction, all gathered approaches, use specific terms in order to deal with e-contacts. Those terms are domain-specific and facilitate dealing with specific open problems. This is the main reason we do not refer in detail the way each approach address the first issue of interest (i.e. ontology). For a more detailed analysis, someone has to study other research approaches which address e-contracting from the perspective of ontologies. This is out of our scope.

3.6 Other Application Domains

3.6.1 Business Process Modeling and Application

In this section we have gathered approaches that see contracts from the enterprise perspective. Specifically, in [15, 16, 52, 49] contracts are used to model and manage enterprise business processes, also known as workflows. As can be observed, the main issue those approaches address is temporal reasoning within business processes, while some of them adopt and represent deontic modalities.

In [15], Davulcu *et al.* propose the Concurrent Transaction Logic (CTR) [6] as a language for specifying, analysing and scheduling of workflows. CTR is a conservative extension of the classical predicate logic and as argued, CTR is capable for (i) representing control flow graphs with transition conditions, (ii) representing triggers, i.e. event-condition-action rules, and is (iii) reasoning temporally. The main idea of this approach is a transformation procedure, called Apply, which accepts a workflow specification, consisting of control flow graphs, triggers and temporal constraints, and constructs an equivalent specification in CTR. In [16] an extension, called CTR-S, is presented. CTR-S extends CTR with certain concepts borrowed from the Game Theory. The problem this approach deals with is adversarial situations that arise in service contracting. A typical case is where contractual parties such as buyers and sellers have conflicting goals. For example, the buyer needs to be assured that goods will either be delivered or money will be returned, while the seller needs to be assured in case of contract break the down-payment can be kept.

Marjanovic and Milosevic, in [52], describe some ideas for e-contract modelling. Formal modelling includes (i) modelling of deontic constraints and verification of deontic consistency, (ii) modelling of temporal constraints and verification of temporal consistency. They use the Reference Model of Open Distributed Processing (RM-ODP) [1] that introduces concepts and terminology to produce an enterprise specification. The basic concepts are: (i) the community, i.e., group of people/agents and resources. Precise behaviour is possible in terms of roles; (ii) the contract that defines obligations, permission and prohibitions. Temporal and deontic constraints are combined to verify temporal and deontic consistency. In this direction, visualization and verification of deontic constraints and their consistency is possible via role windows, while verification of deontic consistency is done through time maps.

The Simple Obligation and Right Model (SORM) is presented by Lubwig and Stolze in [49]. SORM provides an abstract and domain independent model for contractual content representation and management of promises denoted in e-contracts. The corner stone of this approach is the notion of promise. Promises are the matter of subject in the electronic contract. Specifically, the party that promises enters an obligation, while the party that receives the promise holds a right. As mentioned in [49], although a Deontic Logic contract representation is suitable for reasoning about promises and consistency checking, it does not tell us how and when to check entailments for a request or when to check promises. Those issues are addressed in this paper. The main objective of the SORM is to provide a model that supports the monitoring of compliance and fulfilment of the contractual obligations. Towards this direction, contractual obligations and respectively contractual rights are distinguished in (i) state obligation and right, that are obligation and right of parties to maintain a particular state, (ii) obligation to perform a certain action and right to have an action performed, and (iii) option obligation and right to act, that are obligation of a party to tolerate an action performed by another party that has the corresponding right. A suitable representation of those obligations and right types is proposed in the SORM framework. Finally, certain operations performed on the set of active obligations and rights are discussed in order to capture the dynamics of the domain. This approach should be seen with respect to previous works, such as [33, 39], where the CrossFlow architecture is presented. CrossFlow is a contract-based framework that supports the dynamic establishment and enactment of a business relationship between two organizations.

3.6.2 Virtual Communities

Here we discuss approaches that see contracts as a way to regulate agent societies [17, 21, 8, 20, 5, 85]. Social contracts are considered as a set of norms, rules, commitments or conventions that coordinate and manage the society behavior. Generally, social contracts are dynamically determined and stipulated by autonomous agents according to their own internal aims and architecture. It is out of our scope to study the problems and specifications of agent societies, thus we examine only what those approaches consider as contracts and the way they use them.

Dellarocas, in [17], presented a system, called Contractual Agent Societies (CAS), where agents may configure themselves and manage their activities through social contracts. Here contracts include beliefs, values, objectives, protocols and policies. Specifically, a social contract is a social commitment, which is agreed and established among agents, and it forms a particular social relationship and, more importantly, it regulates agents' behavior [9, 42, 79]. An important part of this approach is the social control system, which is responsible for avoiding, detecting and resolving deviations from ideal behavior via incentives (positive or negative sanctions) and sentinels (commitment monitors).

In [21], Dignum *et al.*, presented a framework for agent societies, called OperA (Organizations per Agents) [20]. It consists of three different models. The interrelations between models are described by means of contracts. Here, two types of contacts are described: (i) social contracts that specify commitments between an agent and the society, and (ii) interaction contracts that specify agreements between individual agents. Note that, in this approach, the notion of the social contract differs from the one presented in Dellarocas [17] where both social and interaction contracts are merged into the social contract notion. Both types of contracts are represented through the Logic for Contract Representation (LCR) language, that is based on the Temporal and Deontic Logic (BTLcont) [19] and the branching-time temporal logic (CTL*) [22]. Based on this logic, formulae are represented as branching structures where nodes represent states and arcs represent events. The logic is extended with special operators to address issues such as (i) what is the agent's view on the consequents of actions, (ii) deontic modalities, e.g. obligations, and their violations, (iii) conditional obligations with deadlines, and (iv) CTD imperatives [20].

In the same spirit, Boella and van der Torre [5], address the problem of regulating societies of agents and agents via contracts. Here, contracts are modeled as legal institutions [68]. Boella and van der Torre present three reasons to argue that although most normative systems identify norms with obligations, permissions and prohibitions, this approach is not efficient for complex normative systems. Thus, they formalize obligations in terms of desires and goals, and constitutive rules as beliefs. Constitutive rules create obligations when a contract is stipulated or when some relevant event happens. This notion is close to the conditional obligations as presented in [21]. In an earlier work, Broersen *et al.* are interested in conflicts arising between an agent's beliefs, obligations, intensions and desires [8]. In this approach they use normal default rules [65] to detect conflicts, and priorities that stand among mental states to accomplish conflict resolution. However, they do not address conflicts in a temporal setting.

Wooldridge and van der Hoek, in [85], investigate the relationship between Alternating-time Temporal Logic (ATL) [2] and Deontic Logic that is the link among ability and obligations. Towards this direction, they introduce a variation of ATL called Normative ATL* (NATL*). In this logic, powers and coalitions of agents are seen through the perspective of a normative system, which is a set of rules that constrain the actions of the agents in the system in certain states. They introduce modal operators for permission and obligation and, more importantly, the show how these operators shall be interpreted in terms of normative ability. NATL* is used to formalize the model of the social contract, i.e. the multi-agent system and the social law. To conclude, the main issue this work focuses is the link of requirements 2 and 3. Furthermore, in their future work section, they argue about the need to examine under the scope of NATL* the CTDs structures. This remark is based on Prakken's and Sergot's [61] argument that many of the CTDs paradoxes can be solved within a temporal perspective.

4 Our Approach

In [26, 27], Giannikis *et al.*, proposed an approach for representing e-contracts as Default Theories (DfT). This work focuses on issues such as:

(i) What state the business exchange is in, that is, given a history of events, what factual information is established and what norms are active for each party. This calls for some kind of formal logic representation in order to address requirements 2, 3, 4 and 9.

(ii) If the history of events is incomplete, or if the agent possesses incomplete domain knowledge, reasoning need to employ assumptions; should more information become available later, rendering some of these assumptions false, any conclusions drawn will need to be retracted. Hence, some kind of hypothetical and nonmonotonic reasoning is required towards the direction of requirement 8.

(iii) Whether normative conflicts arise for the agent, that is, whether it finds itself in a situation where it bears norms that it cannot fulfill simultaneously. In such cases, the agent needs some way to manage normative conflicts as described in requirement 7.

To establish the state of the business exchange, given the actions that parties perform or omit to perform, we employ a representation of the agreement in Event Calculus [47]. However, a representation in Event Calculus does not facilitate reasoning with partial or incomplete knowledge, and finally, it does not help towards conflict resolution.

In order to support reasoning with incomplete knowledge one might use some of the various approaches such as the Closed World Assumption [64] or Circumscription [53] or Logic Programs [24, 25] or Defeasible Logic [57] as many researchers have done and mentioned in the previous section. In what follows we investigate the ability of those approaches to reason within domains with incomplete knowledge and record facts and our conclusions that lead us to the use of Reiter's Default Logic (DfL).

In the first case, under the Closed World Assumption (CWA) [64], an atomic formula is assumed false, unless it is known to be true. The agent using a (possibly incomplete) EC contract representation essentially admits into its knowledge base negative literals that correspond to assumptions it makes under CWA, about the falsity of certain atomic formulae. In many realistic scenarios, however, the agent will need to make assumptions about the truth of certain atomic formulae and, furthermore, not only about negative formulae but also about positive ones. Moreover, in this case conflict resolution is not addressed at all.

In the second case, we might consider the EC contact representation as a (General) Logic Program (stable model semantics [24]) or as an Extended Logic Program (answer sets [25]). [34] and [59] in this direction as regards their approach in addressing nonmonotonic reasoning. Under the stable model semantics we observe that entailment is goal-driven i.e., given a logic program LP we define its reduction LP_M w.r.t a set of atoms M. Moreover, the removal of all rules that contain negative literals $\neg B$ (where $B \in M$) and all negative literals in the bodies of the remaining rules presupposes a universal acceptance of assumptions. But, a more important fact is that there is no way to relate the removed literals with the entailment of possible conclusions. Furthermore, as in CWA, no assumptions on positive literals are possible and no specific interpretation is defined when new knowledge is available.

In the third case, we might use Circumscription [53], a generalization of the CWA. Here, we use special predicates to denote abnormal (unexpected) events and effects of actions, and the inference strategy attempts to minimize abnormality. The agent essentially admits into its knowledge base explicit information about abnormality and the conclusions derived are those contained in the minimal models of the augmented knowledge base. [90] is in this direction. Although this approach behaves dynamically, it presents some problems for realistic scenarios: First, it requires that we define abnormal events, effects of actions and the like, explicitly, and, also, that we explicitly distinguish each abnormality from other individuals. Second, in order to decide which individuals to characterize as abnormal, we are required to anticipate the conclusions that we want to be able to derive.

Finally, there is another approach to support default reasoning with e-contracts. [30] and [59] use Defeasible Logic [57]. [30] do not, however, employ some underlying temporal logic. Defeasible Logic allows us to define which conclusions are retractable by making a distinction between strict and defeasible rules. Also it determines the way new knowledge interacts with previous conclusions. The question that arises is that is it possible, on an a priori basis, to determine during the construction of the rule base what is defeasible and what is not? May be in some situations, such as the examples shown in [30] and [59], we are able to determine the type of the applied discount based on the type of the customer (premium or not). But when it comes to assumptions we are not able to know what will be the case. Thus, we believe that in realistic scenarios reasoning dynamically is essential.

Towards this direction, we propose a representation of e-contracts as default theories in Reiter's Default Logic [65]. This contract representation in DfL results as the outcome of the reconstruction of the e-contract's initial EC representation. Specifically, each formula of the EC representation is mapped onto a default rule: the conclusion of the EC formula is mapped to the consequent part of the default rule. Each of the antecedents of the EC formula is mapped either to the prerequisite or to the justification part of the default rule, depending on what information is available in the knowledge base about a particular domain. Knowledge that we want to be proved from the knowledge base is mapped to the prerequisites part of the default rule, while knowledge that is absent from the knowledge base and may be assumed is mapped to the justifications part of the default rule. The formal characterization of the construction of the Default Theory is given in [27].

As regards conflict management, although the EC contract representation facilitates conflict detection it does not facilitates conflict resolution. To this scope, it can be extended with appropriate rules for static priority ascription that determine criteria for conflict resolution. In the same spirit, the other approaches, such as [34, 30, 59] use explicit priority assignment, while [90] does not address conflict management at all. However, realistically, an agent might want to change its criteria over time, to adapt its conflict resolution strategy.

In [26, 28] we address the issues of conflicts detection and dynamic conflict resolution. Specifically, the analysis, representation and management of normative conflicts have been the focus of [28]. In this work we are concerned with normative conflicts that arise for agents engaging in electronic contracting, and present a set of primitive conflict patterns. We also identify some patterns of normative conflict that have not been identified in other proposals. Finally, we argue that the representation of contractual norms as default rules facilitates both conflict detection and resolution.

A DfL representation allows us to detect conflicts by examining extensions, which are essentially sets of propositions. In general, a potential conflict arises when there are multiple extensions of a DfT that represents a contract, and one of them contains a proposition that conflicts with a proposition contained in another, the so called *inter-extension* conflicts [26]. Conflicts may also arise even when there is a single extension of the DfT, if it contains conflicting propositions, the so called *intra-extension* conflicts [26]. The detection of inter-extension conflicts is useful for an agent, which finds itself in a state that is not, *yet*, problematic, and has alternative courses of action to consider. The agent must decide upon a specific course of action – some way of *preventing* the potential conflicts from ever arising is required. The detection of intra-extension conflicts, on the other hand, essentially informs the agent that it is, *already*, in a problematic state. Again the agent needs a way to *resolve* the conflict and decide which norm to satisfy in a way that minimizes the damage done – since, unavoidably, some norm will be violated.

Conflict resolution in DfL may be performed using Brewka's [7] proposal that enables us to define and apply priorities on default rules. What makes PDfTs really useful is that the ascription of priorities to default rules may, itself, be done dynamically. Using dynamic priorities, we generate preferred extensions, each of which indicates a transaction plan. Priorities amongst ground defaults may be defined dynamically either by making different assumptions or by specifying domain-dependent criteria. In this manner, we may manage conflicts in a variety of ways, by specifying different criteria, such as hierarchies of entities of interest, time or specificity of norms.

5 Conclusions

In this report, we presented a survey and classification of logic-based approaches for contract representation and performance monitoring. Through this attempt, besides the critical overview, we have also derived and recorded requirements that a tool for e-contacting should attend. Finally, we discussed some of those approaches with respect to hypothetical reasoning and conflict management and presented our approach to address issues such as temporal defeasible reasoning and conflict detection and resolution.

Acknowledgments

This work was supported by the European Social Fund (75%) and the Greek Secretariat for Research and Technology (25%) through the 3rd Community Support Programme - Measure 8.3 (PENED2003-03EΔ466).

References

[1] ISO/IEC WD 15414, Open Distributed Processing - Reference Model - Enterprise Viewpoint. 1998.

[2] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.

[3] Nick Bassiliades, Grigoris Antoniou, and Ioannis P. Vlahavas. DR-Device: A defeasible logic system for the semantic web. In Hans Jörgen Ohlbach and Sebastian Schaffert, editors, *PPSWR*, volume 3208 of *Lecture Notes in Computer Science*, pages 134–148. Springer, 2004.

[4] Sumit Bhansali and Benjamin N. Grosof. Extending the SweetDeal approach for e-Procurement using SweetRules and RuleML. In Asaf Adi, Suzette Stouten-

burg, and Said Tabet, editors, *RuleML*, volume 3791 of *Lecture Notes in Computer Science*, pages 113–129. Springer, 2005.

[5] Guido Boella and Leendert W. N. van der Torre. Contracts as legal institutions in organizations of autonomous agents. In *AAMAS*, pages 948–955. IEEE Computer Society, 2004.

[6] Anthony J. Bonner and Michael Kifer. Concurrency and communication in transaction logic. In *Joint International Conference and Symposium on Logic Programming*, pages 142–156, 1996.

[7] Gerhard Brewka. Reasoning about priorities in default logic. In *AAAI'94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 2)*, pages 940–945, Menlo Park, CA, USA, 1994. American Association for Artificial Intelligence.

[8] Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zisheng Huang, and Leendert der van Torre. The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In Jφrg P. Möller, Elisabeth Andre, Sandip Sen, and Claude Frasson, editors, *AGENTS01: Proceedings of the Fifth International Conference on Autonomous Agents*, pages 9–16, Montreal, Canada, 2001. ACM Press.

[9] Cristiano Castelfranchi. Commitments: From individual intentions to groups and organizations. In Victor R. Lesser and Les Gasser, editors, *ICMAS*, pages 41–48. The MIT Press, 1995.

[10] Robert L. Causey. EVID: a system for interactive defeasible reasoning. *Decis. Support Syst.*, 11(2):103–131, 1994.

[11] Brian F. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, 1980.

[12] Roderick M. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24:33–36, 1963.

[13] Aspassia Daskalopulu. Modeling legal contracts as processes. In *Legal Information Systems Applications, 11th International Workshop on Database and Expert Systems Applications (DEXA'00)*, pages 1074–1079. IEEE Computer Society, 2000.

[14] Aspassia Daskalopulu and T. S. E. Maibaum. Towards electronic contract performance. In A. Min Tjoa and Roland Wagner, editors, *DEXA Workshop*, pages 771–777. IEEE Computer Society, 2001.

[15] Hasan Davulcu, Michael Kifer, C. R. Ramakrishnan, and I. V. Ramakrishnan. Logic based modeling and analysis of workflows. In *PODS '98: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 25–33, New York, NY, USA, 1998. ACM Press.

[16] Hasan Davulcu, Michael Kifer, and I. V. Ramakrishnan. CTR-S: a logic for specifying contracts in semantic web services. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 144–153, New York, NY, USA, 2004. ACM Press.

[17] Chrysanthos Dellarocas. Contractual agent societies: Negotiated shared context and social control in open multi-agent systems. *Workshop on Norms and Institu-*

tions in Multi-Agent Systems, 4th International Conference on Multi-Agent Systems (Agents-2000), 2000.

[18] Sandra K. Dewitz, Young Ryu, and Ronald M. Lee. Defeasible reasoning in law. *Decis. Support Syst.*, 11(2):133–155, 1994.

[19] F. Dignum and R. Kuiper. Specifying deadlines with dense time using deontic and temporal logic.

[20] Virginia Dignum and Frank Dignum. Modelling agent societies: Co-ordination frameworks and institutions. In Pavel Brazdil and Alpio Jorge, editors, *IPIA*, volume 2258 of *Lecture Notes in Computer Science*, pages 191–204. Springer, 2001.

[21] Virginia Dignum, John-Jules Meyer, Hans Weigand, and Frank Dignum. An organizational-oriented model for agent societies. In *Proceedings of International Workshop on Regulated Agent-Based Social Systems: Theories and Applications. AAMAS'02, Bologna*, 2002.

[22] E. Allen Emerson. Temporal and modal logic. pages 995–1072, 1990.

[23] Andrew D. H. Farrell, Marek J. Sergot, Mathias Sallé, and Claudio Bartolini. Using the event calculus for tracking the normative state of contracts. *Int. J. Cooperative Inf. Syst.*, 14(2-3):99–129, 2005.

[24] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. pages 1070–1080, 1988.

[25] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Comput.*, 9(3/4):365–386, 1991.

[26] Georgios K. Giannikis and Aspassia Daskalopulu. Defeasible reasoning with e-contracts. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2006*, pages 690–694, Hong Kong, China, 2006. IEEE Computer Society.

[27] Georgios K. Giannikis and Aspassia Daskalopulu. The represenation of e-Contracts as default theories. In *Proceedings of 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2007, accepted for publication*, 2007.

[28] Georgios K. Giannikis and Aspassia Daskalopulu. Normative conflict management with default logic. Technical Report A006.3ΓΙΑ, Department of Computer and Communications Engineering, University of Thessaly, A006.3ΓΙΑ, February 2007.

[29] Guido Governatori. Representing business contracts in RuleML. *Int. J. Cooperative Inf. Syst.*, 14(2-3):181–216, 2005.

[30] Guido Governatori and Duy Hoang. A semantic web based architecture for e-contracts in defeasible logic. In *1st International Conference on Rules and Rule Markup Languages for the Semantic Web*, pages 145–159, 2005.

[31] Guido Governatori, Antonino Rotolo, and Giovanni Sartor. Temporalised normative positions in defeasible logic. In *ICAIL*, pages 25–34. ACM, 2005.

[32] Guido Governatori and Antonio Rotolo. Logic of violations: A gentzen system for reasoning with contrary-to-duty obligations. *The Australasian Journal of Logic*, 4:193–215, 2006.

[33] Paul W. P. J. Grefen, Karl Aberer, Heiko Ludwig, and Yigal Hoffner. Cross-Flow: Cross-organizational workflow management for service outsourcing in dynamic virtual enterprises. *IEEE Data Eng. Bull.*, 24(1):52–57, 2001.

[34] Benjamin N. Grosof. Representing e-commerce rules via situated courteous logic programs in RuleML. *Electronic Commerce Research and Applications*, 3(1):2–20, 2004.

[35] Benjamin N. Grosof, Yannis Labrou, and Hoi Y. Chan. A declarative approach to business rules in contracts: courteous logic programs in xml. In *ACM Conference on Electronic Commerce*, pages 68–77, 1999.

[36] Benjamin N. Grosof and Terrence C. Poon. SweetDeal: representing agent contracts with exceptions using XML rules, ontologies, and process descriptions. In *WWW*, pages 340–349, 2003.

[37] Jaap Hage. Law and defeasibility. *Artif. Intell. Law*, 11(2-3):221–243, 2003.

[38] H. Herrestad and C. Krogh. Deontic logic relativised to bearers and counterparties, 1995.

[39] Yigal Hoffner, Simon Field, Paul W. P. J. Grefen, and Heiko Ludwig. Contract-driven creation and operation of virtual enterprises. *Computer Networks*, 37(2):111–136, 2001.

[40] John F. Horty. Moral dilemmas and nonmonotonic logic. *Journal of Philosophical Logic*, 23(1):35–65, 1994.

[41] John F. Horty. Reasoning with moral conflicts. *Nous*, 37(4):557–605, 2003.

[42] N. R. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250, 1993.

[43] Andrew J.I. Jones and Marek J. Sergot. Deontic logic in the representation of law: Towards a methodology. *Artificial Intelligence and Law*, 1(1):45–64, 1992.

[44] Andrew J.I. Jones and Marek J. Sergot. A formal characterisation of institutionalised power. *Journal of the IGPL*, 4(3):427–443, 1996.

[45] Steven O. Kimbrough. Formal language for business communication: sketch of a basic theory. *Int. J. Electron. Commerce*, 3(2):23–44, 1998.

[46] John Knottenbelt and Keith Clark. Contract-related agents. In Francesca Toni and Paolo Torroni, editors, *CLIMA VI*, volume 3900 of *Lecture Notes in Computer Science*, pages 226–242. Springer, 2005.

[47] Robert A. Kowalski and Marek J. Sergot. A logic-based calculus of events. *New Generation Comput.*, 4(1):67–95, 1986.

[48] Ioan Alfred Letia and Adrian Groza. Running contracts with defeasible commitment. In Moonis Ali and Richard Dapoigny, editors, *IEA/AIE*, volume 4031 of *Lecture Notes in Computer Science*, pages 91–100. Springer, 2006.

[49] Heiko Ludwig and Markus Stolze. Simple obligation and right model (SORM) - for the runtime management of electronic service contracts. *Second International Workshop on Web Services, E-Business and the Semantic Web, WES 2003*, pages 62–76, 2003.

[50] David Makinson. On the formal representation of rights relations. *Journal of Philosophical Logic*, 15(4):403–425, 1986.

[51] Rafaël Hernández Marín and Giovanni Sartor. Time and norms: a formalisation in the event-calculus. In *7th International Conference on Artificial Intelligence and Law*, pages 90–99, New York, NY, USA, 1999. ACM Press.

[52] Olivera Marjanovic and Zoran Milosevic. Towards formal modeling of e-contracts. In *EDOC*, pages 59–68. IEEE Computer Society, 2001.

[53] John McCarthy. Circumscription - a form of non-monotonic reasoning. *Artif. Intell.*, 13(1-2):27–39, 1980.

[54] J. J. Ch. Meyer. A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic*, 29(1):109–136, 1988.

[55] J.-J. Ch. Meyer and R. J. Wieringa. Deontic logic: a concise overview. pages 3–16, 1993.

[56] J.-J. Ch. Meyer, R. J. Wieringa, and F. P. M. Dignum. The role of deontic logic in the specification of information systems. pages 71–115, 1998.

[57] Donald Nute. Defeasible logic. In Dov Gabbay, Christopher J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Nonmonotonic Reasoning and Uncertain Reasoning*, volume 3, pages 353–395. Oxford University Press, 1994.

[58] Adrian Paschke and Martin Bichler. SLA representation, management and enforcement. In *EEE*, pages 158–163. IEEE Computer Society, 2005.

[59] Adrian Paschke, Martin Bichler, and Jens Dietrich. ContractLog: An approach to rule based monitoring and execution of service level agreements. In *1st International Conference on Rules and Rule Markup Languages for the Semantic Web*, pages 209–217, 2005.

[60] Henry Prakken and Giovanni Sartor. The three faces of defeasibility in the law. *Ratio Juris*, 17:118–139, 2004.

[61] Henry Prakken and Marek J. Sergot. Contrary-to-duty obligations. *Studia Logica*, 57(1):91–115, 1996.

[62] Anand S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In Rudy van Hoe, editor, *Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Eindhoven, The Netherlands, 1996.

[63] D. Reeves, B. Grosof, M. Wellman, and H. Chan. Towards a declarative language for negotiating executable contracts. In *Proc. AAAI-99 Workshop on Artificial Intelligence in Electronic Commerce (AIEC-99)*.

[64] Raymond Reiter. On closed world data bases. In *Logic and Data Bases*, pages 55–76, 1977.

[65] Raymond Reiter. A logic for default reasoning. *Artif. Intell.*, 13(1-2):81–132, 1980.

[66] Mohsen Rouached, Olivier Perrin, and Claude Godart. A contract-based approach for monitoring collaborative web services using commitments in the event calculus. In Anne H. H. Ngu, Masaru Kitsuregawa, Erich J. Neuhold, Jen-Yao Chung, and Quan Z. Sheng, editors, *6th International Conference on Web Information Systems Engineering*, volume 3806 of *Lecture Notes in Computer Science*, pages 426–434. Springer, 2005.

[67] Mohsen Rouached, Olivier Perrin, and Claude Godart. A contract layered architecture for regulating cross-organisational business processes. In Wil M. P. van der Aalst, Boualem Benatallah, Fabio Casati, and Francisco Curbera, editors, *Business Process Management*, volume 3649, pages 410–415, 2005.

[68] Dick W. P. Ruiter. A basic classification of legal institutions. *Ratio Juris*, 10(4):357–371, 1977.

[69] M. Sallè and A. Boulmakoul. Integrated contract management, 2002.

[70] Mathias Sallè. Electronic contract framework: An overview. *AAAI-2002 Workshop on AgentTechnologies for B2B Electronic Commerce*, 2002.

[71] Mathias Sallè. Electronic contract framework for contractual agents. In Robin Cohen and Bruce Spencer, editors, *Canadian Conference on AI*, volume 2338 of *Lecture Notes in Computer Science*, pages 349–353. Springer, 2002.

[72] F. Santos and J. Carmo. A deontic logic representation of contractual obligations. pages 243–257, 1994.

[73] Filipe Santos and Josè Carmo. Indirect action, influence and responsibility. In Mark A. Brown and Josı Carmo, editors, *DEON*, Workshops in Computing, pages 194–215. Springer, 1996.

[74] Filipe A. A. Santos, Andrew J. I. Jones, and Josı Carmo. Action concepts for describing organised interaction. In *HICSS (5)*, pages 373–382, 1997.

[75] Giovanni Sartor. Normative conflicts in legal reasoning. *Artificial Intelligence and Law*, 1(2-3):209–235, 1992.

[76] Giovanni Sartor. A simple computational model for nonmonotonic and adversarial legal reasoning. In *ICAIL '93: Proceedings of the 4th international conference on Artificial intelligence and law*, pages 192–201, 1993.

[77] Murray Shanahan. The event calculus explained. In *Artificial Intelligence Today*, pages 409–430. 1999.

[78] Murray Shanahan. Review: M. Shanahan, Solving the frame problem. *Artif. Intell.*, 123(1-2):275, 2000.

[79] Munindar P. Singh. An ontology for commitments in multiagent systems. *Artif. Intell. Law*, 7(1):97–113, 1999.

[80] Yao-Hua Tan and Walter Thoen. Modeling directed obligations and permissions in trade contracts. In *HICSS (5)*, pages 166–175, 1998.

[81] Yao-Hua Tan and Walter Thoen. Using event semantics for modeling contracts. In *HICSS*, page 170, 2002.

[82] Lluis Vila and Hajime Yoshino. Time in automated legal reasoning. Technical Report ICS-TR-96-57, 1996.

[83] Georg H. von Wright. Deontic logic. *Mind, Oxford University Press*, 60:1–15, 1951.

[84] R. J. Wieringa and J.-J. Ch. Meyer. Applications of deontic logic in computer science: a concise overview. pages 17–40, 1993.

[85] Michael Wooldridge and Wiebe van der Hoek. On obligations and normative ability: Towards a logical analysis of the social contract. *J. Applied Logic*, 3(3-4):396–420, 2005.

[86] Lai Xu. A framework for e-markets: Monitoring contract fulfillment. In Christoph Bussler, Dieter Fensel, Maria E. Orlowska, and Jian Yang, editors, *WES*, volume 3095 of *Lecture Notes in Computer Science*, pages 51–61. Springer, 2003.

[87] Lai Xu. Monitorable electronic contract. In *CEC*, pages 92–. IEEE Computer Society, 2003.

[88] Lai Xu and Manfred A. Jeusfeld. Pro-active monitoring of electronic contracts. In Johann Eder and Michele Missikoff, editors, *CAiSE*, volume 2681 of *Lecture Notes in Computer Science*, pages 584–600. Springer, 2003.

[89] Lai Xu and Manfred A. Jeusfeld. Detecting violators of multi-party contracts. In Robert Meersman and Zahir Tari, editors, *CoopIS/DOA/ODBASE (1)*, volume 3290 of *Lecture Notes in Computer Science*, pages 526–543. Springer, 2004.

[90] Pinar Yolum and Munindar P. Singh. Reasoning about commitments in the event calculus: An approach for specifying and executing protocols. *Ann. Math. Artif. Intell.*, 42(1-3):227–253, 2004.