# Automated eContract Negotiation in Web Service Environment: Electronic Contract Management Aspects

Marius ŠAUČIŪNAS
*Institute of Mathematics and Informatics, Vilnius University*
*Akademijos Str. 4, Vilnius, Lithuania*
*m.sauciunas@gmail.com*

**Abstract.** The paper addresses the electronic contract management problems in automated eContract negotiation among software agents in the web service environment. From the point of electronic contract management, the aim of negotiation process is to automatically form contractual agreements between different parties, coordinating their behavior and facilitating contract execution. The contracts specify the commitments that the involved parties make to each other and that play the important role in their interactions. Therefore in the contract representation language the commitments must be explicitly represented and specify what should be done, if the legal norms defined by policies are violated, and inform the parties about the behavior they could expect from the others. The paper familiarizes with the details of electronic contract representation problem and with approaches which has been proposed to solve this problem. It presents also a critical analysis of the proposed approaches and summarizes their challenges and drawbacks. The paper analyses also one of more advanced conceptual framework of negotiation process from the electronic contract representation perspective, highlights its drawbacks and proposes how to improve this framework.

**Keywords.** eContract, contract representation language, deontic logic

## Introduction

The subject of this paper is the critical analysis of the electronic contract management process among the software agents in the web service environment. At present time the whole contract lifecycle in eBusiness, including the negotiation, preparation of eContract and its acceptation, predominantly is handled manually. In order to develop an electronic contract, humans should not only write and agree upon it but also to translate manually into some computer-readable internal representation [6]. The negotiation, preparation and usage of eContracts still is a challenge.

The electronic contract representation is especially important in the dynamic environments in which prevail the short time contracts. Such contracts have to be dynamically set to meet end-users and service providers' short period needs. In such circumstances, contracts have an intrinsic dynamic and flexible nature and have to regulate independent behavior of diverse parties. Electronic contract preparation and execution facilitation of is one of central issues in the eContracts area.

The goal of the paper is to discuss state of art of the electronic contract representation problem, to highlight the challenges and the drawbacks of the proposed solutions, and to contrasts the conceptual modeling problems of the electronic contract management aspects with the current negotiation process modeling concepts. The main contribution of the paper is the proposal how to improve one of more advanced negotiation process object-oriented modeling framework [11].

## 1. Electronic Contract Representation Problem

Electronic contract representation problem arises in the context of eContracting. One of the most important requirements in the semantic Web environments that the eContracts should prepared automatically, evaluated, negotiated and executed without human intervention when electronic contract have an intrinsic dynamic and flexible nature and have to regulate independent behavior of diverse parties. It seems that the representation of electronic contracts is one of central issues in the electronic contract monitoring and evaluation too. As pointed in [12], even the most of research in this area focuses on this problem. The contracts specify the commitments that the involved parties make to each other and that play the important role in their interactions. The contract is the statement of intent that regulates the behavior of involved organizations and individuals. Therefore in the contract representation language the commitments must be explicitly represented and specify what should be done, if the legal norms defined by policies are violated, and inform the parties about the behaviour they could expect from the others. A number of such languages – LRC [3], DocLog [16], SweetDeal [5], CEL [17], BCL [13], eContracts [10], etc. – have been proposed.

## 2. Electronic Contract Representation Languages

LCR (Logic for Contract Representation) is a language for description interaction in multi-agent systems. This language based on branching-time logic, i.e. the formulae in LCR are interpreted over tree-type branching structures that represent all conceivable ways the system can evolve [3]. The formalism behind the language extends branching-time temporal logic with the deontic relations. In LCR, contract clauses are represented as deontic expressions. The violations and sanctions can also be defined in LCR. However, the main purpose of the language is to formalize the behavior of multi-agent system and to relate this behavior to the global objectives of the system. LCR is not intended to be used in the web service environment.

DocLog [16] is an XML based representation language for contract terms. It is based also on the principles of deontic and action logic. Contractual obligations are treated as norms and represented in a semi-formal way using the extended norm frames [9]. Although the DocLog is intended to be used in the eComerce environment and to support the contract negotiation, the language cannot be used in sophisticated Semantic Web environment because it cannot represent exceptions, temporal and some other important aspects of contracts, and is semi-formal.

Quiet different approach is used in the SweetDeal [1][5] approach. It is a rule-based approach for e-commerce business contracts representation. To describe and communicate contracts, the SweetDeal uses a modular set of logic programming rules about agent contracts with exception handling on top of descriptive logic based

ontologies, which describe business processes. Such approach enables software agents to automate the creation, negotiation, execution, evaluation of the contracts and reuse contract description for multiple purposes. The motivation for rule-based approach is that rules as knowledge representation formalism is relatively mature, suitable for prescriptive specification and already long time ago integrated into software engineering mainstream techniques. The advantages of rules for representing executables electronic contracts are that rules are relatively easy to modify dynamically and that they are a high abstraction level formalism, which, at least theoretically, is closer to humans' understandability [14].

The start point of the SweetDeal formalism was pure logic programs. Over the pure logic programs the so-called acyclic (non-recursive) courteous logic programs (CLP) have been defined. CLP is a superclass of ordinary logic programs. It is equipped with classical negation and prioritized conflict handling mechanisms [4]. Classical negation is permitted in rule heads and bodies. The courteous approach is a hybrid approach that integrates the ideas of logic programming and general non-monotonic reasoning. The procedural attachments can be attached to logical conditions in the rule antecedents and consequents. CLP with procedural attachments are called situated CLP (SCLP). SCLP expressively extends declarative ordinary logic programs to include prioritized conflict handling. This enables modularity in specifying and revising rule-sets. In the SweetDeal formalism the rules are represented as XML documents [4]. Such representation enhances human readability, supports inclusion and generation of textual information and facilitates parsing. Later this knowledge representation formalism evolved into RuleML family [5]. RuleML was proposed as an alternative to SWRL standard [7]. In summary, a SCLP is suitable to represent fully-specified executable contracts as well as partially-specified contracts that are in the midst of being negotiated [14]. The partially-specified contracts can be viewed as contract templates [14]. The set of negotiables and the structure of a contract in terms of services and attributes are specified by process, contract and other ontologies. Syntactically, the names of predicates appearing in the rules may denote classes and properties in OWL ontology and the names of individuals appearing in rules may refer to individuals in an appropriate ontology. Semantically, the referenced ontological knowledge base is viewed as a background theory for the rule base. Inter allies, ontological knowledge is used for exception handling. During the execution of the contract exception condition (e.g. late delivery, non-payment) could occur, for handling these exceptions process knowledge is required. Thus, ontologies enable to specify more complex contracts with behavioral provisions [5].

The SweetDeal approach is supported by an integrated set of tools, SweetRules [15], that supports creation, evaluation, negotiation, execution and monitoring of formal e- contracts. It provides also a communication protocol between the contracting agents, contract knowledge bases and agent communication knowledge bases. Contract negotiation messages exchanged between the parties are considered as contract knowledge bases that are executable in the SweetRules environment. Each knowledge base consists of six parts: rules, facts, ontologies, effectors, fact-queries, their answers, and conditional queries [1]. Rules describe if-then implications of contractual fragments. Facts are rules without bodies. Ontologies define vocabularies over which the rules are defined. OWL ontologies and rule-based object-oriented default inheritance ontologies are allowed. Effectors are procedural attachments of SLCP. They can execute real-world business process (e.g. e-mail messaging) that are

associated with the execution of the contract. Each agent (i.e. negotiating party) has internal knowledge base containing rules that facilitate its communication [1].

In summary, the SweetDeal approach is an well-theoretically-grounded approach that supports many aspects of e-contracting and negotiation. However it does not provide any means to describe deontic modalities and, consequently, is not sufficient to define all legal aspects of negotiated contracts.

Due to volume limitations of this paper other languages will not be discussed in detail.

## 3. Conceptual Modeling of Negotiation Process and Electronic Contract Management Aspects

### *3.1. Analysis of the Lin's Conceptual Framework from the Electronic Contract Representation Perspective*

The Lin's conceptual framework [11] is one of wide-accepted conceptual models of the negotiation process for Web services contracting. He sees this process as a collaboration of the three conceptual entities: the service requester, the service provider and the service discovery agency.

In the semantic Web environments the eContracts should prepared automatically, evaluated, negotiated and executed without human intervention by software agents. In order to negotiate about contractual agreements, the conceptual model should provide mechanisms to specify contract structure and content, related to contract representation, normative statements, related to involved parties behavior regulation and semantic meaning, related to meaning of contract concepts provision. Lin's conceptual framework [11] does not provide any details how to do this. Most problematic issues are the way in which the framework models the contract manager, and the proposed protocol rules, for the signing of the contract under designated contract template. The model assumes that the contract manager, component of service requester, is maintaining contract template for making agreements. In eContracting environment, where contractual agreement has an intrinsic dynamic and flexible nature, they should be managed and negotiated by several different parties with different features and characteristics. Contracts can never be static, rigid and agreed always under the same contract template unless all template terms which are required for all situations will be defined, but this will be quite difficult to manage for Semantic Web. Another way of contract template using is to combine this solution to some other which enables to form the architecture for automatic contract negotiation. Negotiation mechanisms which have to be used for negotiable parameters of the contract template and to get the final contract in not described either. Contract template structure and content not defined too. Besides, the assumption that the service requester must maintain contract template is questionable. These are obvious drawbacks from the contract management perspective. Another drawback is that parties of automated contract negotiation process have no common understanding on the concepts they agree, i.e. proposed model not deal with semantic meaning on the party's used terminology. Besides supporting of contracts semantics is significant on achieving domain independency in semantic web environments. Another drawback is that the model does not provide any business process monitoring solution. One more assumption in the model that the agency collects the evaluations of service providers' presented by the requesters (trust values in

terms of the author) and this information should be enough to evaluate the service, nevertheless for sophisticated evaluation according to agreed contract details this information is insufficient.

### *3.2. Propositions how to Improve Lin's Conceptual Framework*

To adapt the Lin's conceptual framework to the needs of electronic contract management issues, it necessary, first of all, to remove the above discussed drawbacks.

*Contract structure and content.* The model do not provide any information regarding contract structure and content, consequently some XML-based languages, designed to express contractual agreements in a form, understandable for human beings, could be used for this aim.

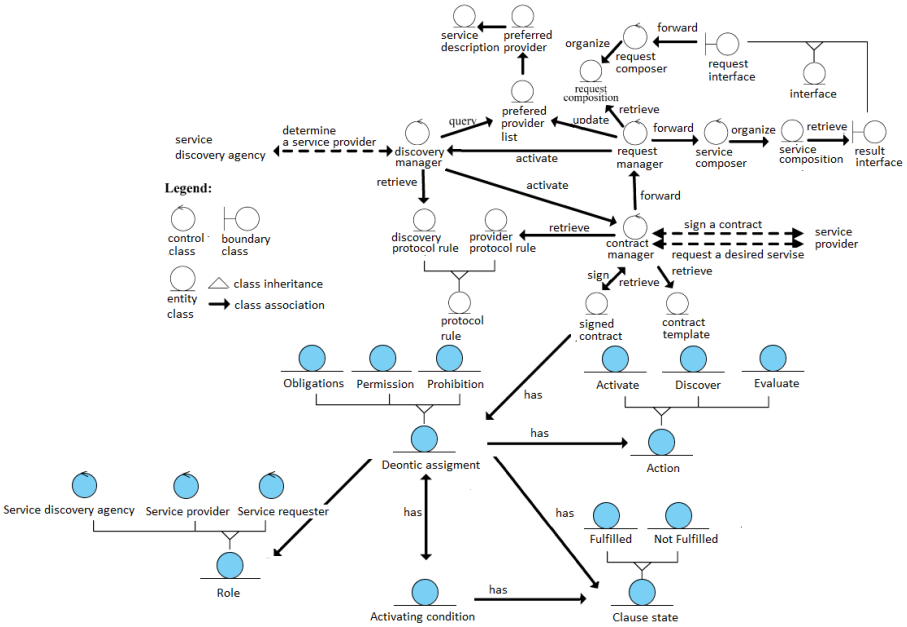


**Figure 1.** Object class diagram extension for the service requester

*Involved parties' behavior regulation and semantic meaning.* One of the most important requirements in the context of eContracting is that in semantic Web environments the eContracts should prepared automatically, without human intervention. Contracts should be prepared and established by software agents. Every contract can be modeled as set of different roles, that allocates the tasks to the agents and set of different clauses that regulates the behavior of them. Every agent, depending on the role it is playing in electronic contract, is able or must to perform certain action. The behavior of contractual agents needs to be regulated after contract establishment. For this purpose contract norms, regulating the behavior, can be specified in electronic contract. These normative statements can be modeled based on deontic logic, the logic of the normative concepts, which represents agent's relationship with the concepts of obligation – agent have to do and action, permission – agent is allowed to do a action, and prohibition – agent isn't allowed to do an action. These concepts could be extended by sanction concept - applied in case certain obligation hasn't been fulfilled. In most

cases these concepts of deontic logic could be used to model electronic contract. Another serious requirement for eContracts that involved parties should have common terminology and interpretation of the contract concepts they agreed on. To achieve it, ontology, which provides common interpretation in the domain, could be used.

Proposed model extension presents how to incorporate common and domain ontologies. Common ontology provides meaning for general terms, needed for every contract, while domain ontology provide domain meaning to the same terms, described in common ontology, but make them domain independent. Common ontology, as mentioned before, describes the general terms of the contracts e.g. deontic assignments, modeled in deontic logic, specifies roles to perform an certain action. Activating condition specifies conditions, which activates deontic assignment, then current state of the statement could be tracked. Domain ontology extends common ontology terms, e.g. Role class can be specified by three subclasses, dominating in Lin's model, the same rule is valid and for Action class. All these propositions modeled and provided as extension to object class diagram (Figure 1.).

*Contract monitoring.* The volume of this paper does not allow me to discuss the required solutions in detail. The main idea is that mechanisms similar to that are provided by CONTRACT [8], TPaML [2] or ECL projects and can be used for this aim.

## 4. Summary and Conclusions

In this paper, the critical analysis of the electronic contract management process among software agents in the web service environment has been performed. In such environment, electronic contracts, with respect to intrinsic dynamic and flexible nature, have to be prepared automatically, evaluated, negotiated and executed without human intervention. The contract is the statement of intent that regulates the behavior of involved organizations and individuals. From the electronic contract management perspective, several significant aspects, such as contract structure, mechanisms to govern collaboration between parties and contract semantic, for specifying complex contracts with behavioral provisions, have to be taken into account dealing with the eContract management problem. From this perspective, major groups of approaches and mechanisms facilitating the electronic contract management problem can be identified: an XML-based languages, languages based on branching-time, deontic, action, programming rules logic, policy-based approaches, role-based approaches. The drawbacks and challenges of each group have been discussed in the paper. Further, the object-oriented Lin's negotiation model [11] that is accepted by many researchers working in the automated negotiation field has been evaluated from the electronic contract management perspective. Its shortcomings have been highlighted, and some significant improvements of the model have been proposed.

The critical analysis of the automated eContract management problem demonstrates, that a lot of different approaches and useful ideas have been proposed up to date, some of them could be distinguished for further investigation and model improvement, e.g. XML-based languages, designed to express contractual agreements in a form, understandable for human beings, role based architectures, contract templates, ontologies, another solutions, not mentioned in this paper due to volume limitation, but which are important to monitoring and evaluation areas. A lot of experimental research should be done for this aim. It intends to be a major focus of my further research.

## References


[1] S. Bhansali and B. N. Grosof. Extending the SweetDeal approach for e-procurement using SweetRules and RuleML. In: A. Adi, S. Stoutenburg, S. Tabet, editors, *Proceedings of the First International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML 2005)*, Galway, Ireland, November 10-12, 2005. Lecture Notes in Computer Science **3791**(2005), Springer, Berlin, 113-129.

[2] A. Dan, D. M. Dias, R. Kearney, T. C. Lau, T. N. Nguyen, F. N. Parr, M. W. Sachs, and H. H. Shaikh. Business-to-business integration with TpaML and a business-to-business protocol framework. *IBM Systems Journal* **40**(1) 2001, 68-90.

[3] V. Dignum, J-J. Meyer, F. Dignum, and H. Weigand. Formal specification of interaction in agent societies. In: M. Hinchey, J. Rash, W. Truszkowski, C. Rouff, and D. Gordon-Spears, editors, *Formal Approaches to Agent-Based Systems (FAABS)*. Lecture Notes in Computer Science **2699** (2003), Springer, Berlin, 37-52.

[4] B. N. Grosof, *Courteous Logic Programs: Prioritized Conflict Handling for Rules*. IBM Research Report RC 20836, December 30 1997, revised from May 8 1997. Available from: http://www.mit.edu/~bgrosof/paps/rc20836.pdf.

[5] B. Grosof, A roadmap for rules and RuleML in the Semantic Web. *IEEE Intelligent Systems* **18**(5) (2003), 76-83. Available from: http://www.mit.edu/~bgrosof/paps/ruleml-ieee-intell-sys-2003.pdf.

[6] P. Hasselmeyer, Ch. Qu, L. Schubert, B. Koller, and Ph. Wieder, Towards autonomous brokered SLA negotiation. In: P. Cunningham, M. Cunningham, editors, *Exploiting the Knowledge Economy: Issues, Applications, Case Studies (eChallenges 2006)*, Barcelona, Spain, October 2006. IOS Press, 44-51. Available from: http://www.hasselmeyer.eu/pdf/echal06.pdf.

[7] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, *SWRL: a Semantic Web Rule Language Combining OWL and RuleML*, 2004. Available from: http://www.w3.org/Submission/SWRL/.

[8] *Contract based Electronic Business Systems State of the Art*. IST Contract Project Deliverable, 2007.

[9] R. van Kralingen, A conceptual frame-based ontology for the law. In: *Proceedings of the First International Workshop on Legal Ontologies*, Melbourne, Australia, 1997, 6-17.

[10] L. Leff and P. Meyer, editors, *eContracts. Version 1.0*. Committee specification. OASIS, 27 April 2007. Available from: http://docs.oasis-open.org/legalxml-econtracts/legalxml-econtracts-specification-1.0.html.

[11] J. Lin, A conceptual model for negotiating in service-oriented environments, *Information Processing Letters* **108**(4) 2008, 192–203.

[12] D. Mobach, *Agent-Based Mediated Service Negotiation*, PhD thesis, Computer Science Department, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands, 2007.

[13] S. Neal, J. Cole, P. F. Linington, Z. Milosevic, S. Gibson, and S. Kulkarni, Identifying requirements for business contract language: a monitoring perspective. In: *Proceedings of the Seventh IEEE International Enterprise Distributed Object Computing Conference (EDOC 2003)*, Brisbane, Australia, 16-19 September 2003. IEEE Computer Society, 2003, 50-61.

[14] D. M. Reeves, B. N. Grosof, M. P. Wellman, and H. Y. Chan, *Towards a Declarative Language for Negotiating Executable Contracts*. IBM Research Report, Computer Science, RC 21476 (96914), 11 May 1999. Available from: https://www.aaai.org/Papers/Workshops/1999/WS-99-01/WS99-01-007.pdf.

[15] *SweetRules: Tools for Semantic Web Rules and Ontologies, including Translation, Inferencing, Analysis, and Authoring*, SweetRules, 2005. Available from: http://sweetrules.projects.semwebcentral.org/.

[16] Y. Tan and W. Thoen, DocLog: an electronic contract representation language. In: *Proceedings of 11th International Workshop on Database and Expert Systems Applications (DEXA'00),* 6-8 September 2000, Greenwich, London, UK. IEEE Computer Society, 2000, 1069-1073. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=875159.

[17] X. Wang, E. Chen, D. Radbel, T. Horioka, J. Clark, and G. Wiley, The contract expression language – CEL. In: *Proceedings of the IEEE Workshop on Contract Architectures and Languages*, 20-24 September 2004, Monterey, California, USA. Available from: http://www.contentguard.com/drmwhitepapers/The_CEL.pdf.