An Agent-based Framework for the Automation of Contractual Relationships

Mathias Sallé
Trusted E-Service Laboratory
HP Laboratories Palo Alto
HPL-2002-133
May 7th , 2002*

E-mail: mathias_salle@hp.com

Electronic Contract, multi agent system, contractual agents

In this paper, we present work in progress on the Electronic Contract Framework (ECF), an agent-based framework for the automation of contractual relationships. Electronic contracts are the cornerstone of this framework and contractual agents manipulate these abstractions to specify business relationships and to automate their execution. We first define the concepts and the formalism used to model electronic contracts. Based on the dynamic nature of these concepts, we introduce a collaborative protocol used by contractual agents to synchronize their views on contractual commitments. Lastly, we present the conceptual architecture and the high-level reasoning process of contractual agents.

* Internal Accession Date Only Approved for External Publication
To be published in and presented at the AAAI-2002 Workshop on Agent-Based Technologies for B2B Electronic Commerce, 28 July 2002, Edmonton, Alberta, Canada

An Agent-based Framework
for the Automation of Contractual Relationships

Mathias Sallé
Hewlett-Packard Laboratories
1501 Page Mill Rd,
Palo Alto, CA94304,
United States
mathias_salle@hp.com

Abstract

In this paper, we present work in progress on the Electronic Contract Framework (ECF), an agent-based framework for the automation of contractual relationships. Electronic contracts are the cornerstone of this framework and contractual agents manipulate these abstractions to specify business relationships and to automate their execution. We first define the concepts and the formalism used to model electronic contracts. Based on the dynamic nature of these concepts, we introduce a collaborative protocol used by contractual agents to synchronize their views on contractual commitments. Lastly, we present the conceptual architecture and the high-level reasoning process of contractual agents.

Introduction

The area of business to business (B2B) in electronic commerce has received an increasing interest over the last few years. Based on abstractions such as public and private business processes, various initiatives (RosettaNet, Ebxml) in the IT industry have developed and deployed systems to enable companies to conduct business over the internet. The focus of these B2B frameworks, also referred to as B2Bi frameworks, is towards the integration and the execution of commonly specified public processes (payment, billing). Although they represent an important step for e-commerce, automation of the various steps of the B2B lifecycle has not yet been achieved. We believe that to reach such goal, systems need to be developed on top of higher abstractions to allow for more expressivity in the specification of business relationships and hence better reasoning capabilities.

By nature, business relationships are unfriendly and cooperation cannot be guaranteed. In such a context, contracts are used to reduce the uncertainty and make the behaviour of other parties much more predictable. Furthermore, research on normative reasoning ((Castelfranchi *et al.* 1999), (Dellarocas 2000), (Kollingbaum & Norman 2001), (Morciniec, Salle, & Monahan 2001)) has shown that agents could evolve in regulated environments and adopt different behaviors based on their perception of the norms specified in electronic contracts. Consequently, the notion of electronic contract appears to be abstract enough and close enough to the realm of B2B to be investigated as a modelling abstraction to achieve the desired level of automation.

The Electronic Contract Framework (ECF) is an agent-based framework layered on top of existing B2Bi frameworks. Its objective is to automate the lifecycle of contractual relationships. As described in (Preist 2001) and illustrated in Figure 1 , contractual agents base each negotiation on a contract template and negotiate over the various contract parameters. As a result of each successful negotiation, agents sign up to a contract specifying the agreed business relationship. Each agent then deploys the agreement within their contractual framework. This results in the contract being turned into an internal structure (runtime contract). The contractual framework then monitors the contract and when appropriate drives the execution of the contractual commitments through its connections with the B2B framework. The contract may then terminate normally or be terminated prematurely.

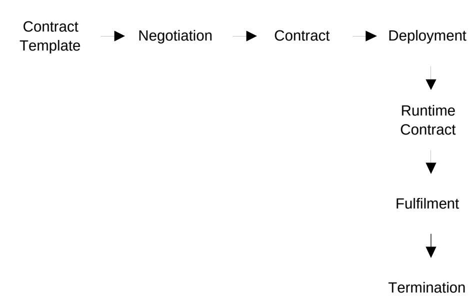

Figure 1: Contractual Relationship Lifecycle

Electronic contracts are the cornerstone of this framework and contractual agents manipulate these abstractions in order to achieve the required degree of automation.

This paper is organized has follows. We first present the concepts and the formalism used to model electronic contracts in ECF. We then introduce the Contract Fulfilment

Protocol and describe how contractual agents can communicate about norms. We then give an overview of the architecture and the high-level reasoning algorithm used for contractual agents. Lastly, we present details on the ECF prototype.

Electronic Contracts

Formalism

Contracts in ECF are defined as statements of intent. They formally specify the behaviour that each contractual party is expected to follow in an ideal world or in sub-ideal situations occurring when one or more parties do not fulfil one or several of their contractual commitments. Contracts are not considered as constraints on behaviour and agents have a deliberative approach when reasoning about them.

A contract is a compound object. It contains an informative section that consist of:

- A contract *identification number* that uniquely identifies the contract for the parties involved.
- The *mappings* between identities and roles.
- The contract *validity period* (start date, expiration date).
- The *normative system* of reference (online or offline institution).

The second section is a behavioral specification. It is a set of normative statements describing the expected behaviour of the various roles defined in the informative section. Normative statements are based on the operators of deontic logic introduced in (von Wright 1950). A formal representation of a normative statement is given below:

$$\text{ns:}\varphi \rightarrow \theta_{s,b}(\alpha < \psi)$$

where:

- ns is a **label** referencing this normative statement.
- φ is the **condition** under which θ obtains.
- θ is a **deontic operator**, Obligation (O), Permission (P) or Prohibition (F).
- s is the **subject** of θ, or the role that assumes θ.
- b is the **beneficiary** of θ, or the role to whom θ is owed.
- α is the **action** to perform or the **state-of-affair** to bring about.
- ψ is a **deadline**

Normative statement are read as follows: "if φ holds then s is {obliged, permitted, prohibited} {to/by} b to achieve α {before/until} ψ holds true".

Non-conditional normative statements are defined as:

$$\text{ns:}\theta_{s,b}(\alpha < \psi)$$

The semantics of the deontic operators are inspired by Meyer's dynamic logic formalism of deontic logic (Meyer Winter 1988). In our framework, obligations are no longer absolute but are relative to their associated sanctions. This extension gives space for deliberative decisions by the agent as whether to fulfil or not a normative statement.

Sanction

Sanctions play an important role in contractual relationships. As Raz [14] argues, sanctions and the desire to avoid them act as a practical reason for agents to comply with norms. Through the definition of sanctions, one gives the means for agents to reason about the consequences of their actions not only in terms of positive consequences but also in terms of negative consequences. Agents consider both the norm and the sanction to decide whether to fulfil the norm or not.

As we investigate the automation of contractual relationships, we want to minimize the number of conflicts resulting in an escalation to the normative system associated with the contract. We refer to this property as *conflict avoidance*. Embedded in this notion of conflict avoidance is the notion of transition from ideal to sub-ideal and the recovery from sub-ideal to ideal situations when possible. As suggested in (van der Torre 1999), if the violation of a clause is not too serious, or was not intended by the violating party, the contracting parties usually do not want to consider this as a breach of contracts but simply as a disruption in the execution of the contract that has to be repaired.

We suggest that there exists at least two classes of sanctions. The first class, referred to as *endogenous sanctions*, are sanctions specified in the contract. Endogenous sanctions are applied in a straight forward manner when violation of the associated clauses occur. Endogenous sanctions are modelled by ns2 in the following example:

ns1:$O_{se,by}(\alpha < \psi_1)$
ns2:$not\_fulfilled(ns1) \rightarrow O_{se,by}(\beta < \psi_2)$

where *not_fulfilled(label)* is a special predicate specifying that an obligation has not been fulfilled.

Endogenous sanctions can be further refined in different categories based on their impact on the contract once they have been applied.

The second class, referred to as *exogenous sanctions*, are sanctions defined by a normative system of reference resulting from a violation of clauses with no specified endogenous sanction or when a sanction is not carried out. Exogenous sanctions can be seen as authoritative determinations of sub-ideal situations by the norm-applying institution of a normative system. A violation of a norm leads to a state of liability to sanction and that state leads to exogenous sanctions defined by the normative system of reference. In a contract, one would specify that not fulfilling an obligation will lead to the state of liability to sanction and the definition of the sanction would be done outside by the normative system as presented below:

ns1:$O_{se,by}(\alpha < \psi_1)$

ns2:$not\_fulfiled(ns1) \rightarrow S$

where S is the state of liability to sanctions. From a contractual point of view, S symbolizes a breach of contract that can not be recovered automatically.

Irrespective of their class (endogenous or exogenous), sanctions are of two types, either:

- The beneficiary of the violated norm is granted a right.

  ns1:$O_{se,by}(deliver(good) < date(order\_time + 30))$
  ns2:$not\_fulfiled(ns1) \rightarrow O_{se,by}(give\_discount(se, by, 25\%))$

  The subject is obliged to deliver a good within 30 minutes of receiving an purchase order. If he fails, the beneficiary is given a 25% discount on next order.

- The subject of the violated norm is refused a right

  ns1:$O_{se,by}(pay(\$500) < \psi_1)$
  ns2:$not\_fulfiled(ns1) \rightarrow F_{se,by}(order\_good(se, by) < pay(se, by, \$500)$

  The subject is obliged to pay $500, if he fails to do so then he is prohibited to order any other goods until he has paid the amount due.

Example

Using normative statements, one can model a procurement contract as follows:

ns1:$O_{se,by}(advance\_notice(s, b) < date(delivery\_date - notice\_period))$

ns2:$O_{se,by}(deliver(se, by, product, quantity) < date(delivery\_date))$

ns3:$fulfilled(ns2) \rightarrow O_{by,se}(pay(by, se, price) < date(delivery\_date + 30))$

ns5:$not\_fulfilled(ns2) \rightarrow O_{se,by}(paypenalty(se, by, percent, price) < date(delivery\_date + 5))$

ns6:$not\_fulfilled(ns3) \rightarrow F_{by,se}(order\_good(by, se) < date(delivery\_date + 30 + 30))$
$\wedge \; O_{by,se}(paypenalty(by, se, percent, total_price, currency) < date(delivery\_date + 30 + 10))$
$\wedge \; O_{by,se}(pay(by, se, quantity, unit\_price, currency) < date(delivery\_date + 30 + 30))$

In this contract, two roles are defined, the buyer (by) and the seller (se). The seller must deliver a given quantity of a product before an agreed delivery date. Furthermore, he must send an advance shipment notice to the buyer a certain number of days (notice_period) before the agreed delivery date. Payments for the given goods by the buyer must be 30 days or less after the delivery date. If the seller fails to deliver the goods on time, the buyer shall pay the seller a penalty fee of an agreed percentage of the order price within 5 days of the agreed delivery date. If the buyer fails to pay the seller on time, the buyer will be prohibited to order goods from the seller for a period of 1 month from the agreed delivery date and the buyer will have to pay a penalty fee of an agreed percentage within 10 days of the agreed original payment deadline and he will have to pay the original amount due within a month of the original payment deadline.

Communicating about norms

Normative Statements Lifecycle

Our analysis revealed that deontic operators are dynamic concepts and it is possible to reason on their dynamic nature. In our model, we associate a well defined lifecycle to each deontic operator. This lifecycle is used in conjunction with the operator's semantics in the reasoning algorithms (compliance checking, etc.). For instance, when a contract is agreed, obligations are in an agreed state, then they move in a pending state when their associated conditions evaluate to true. After, they may move to a fulfilled state if their associated actions are executed. The lifecycle of the operators describes the different states in which they can be and how those states relate to each others. Compliance checking is achieved through the monitoring of the commitments lifecycle and comparison between the expected behaviour and the actual behaviour of the contractual parties.

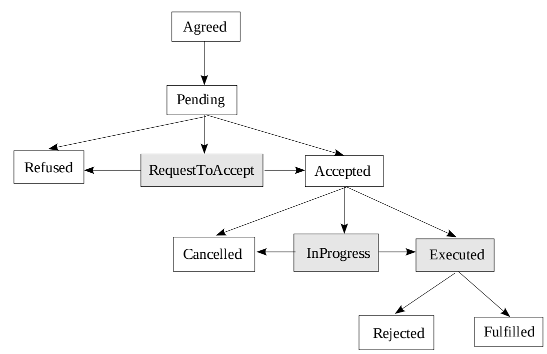

Figure 2: Obligation Lifecycle

We have identified two types of states in the lifecycle of deontic operators. The first one, referred to as *internal states* (shaded states in Figure 2) are internal views on the deontic operator. The second type, referred to as *shared states* (white states in Figure 2) are views on the deontic operator shared by each party involved. Agents need to synchronize their views on these shared states.

Contract Fulfilment Protocol

By nature, contractual relationships are highly distributed. This introduces differences in the view that each contractual agent has on the contractual commitments. For instance, given the following contractual commitment:

$O_{by,se}(pay(\$500) < date(15/03/02))$

When the buyer transfers $500 to the sellers bank account, he will believe that he has fulfilled his obligation. However, it might take three days for this amount to appear in the seller's account. During that time, the seller will still believe that the buyer has to fulfil his obligation. It is important to reconcile these views in order to make sure that each agent shares the same view on the contract. We refer to this as *synchronization*.

As contracts define expected behaviors, it is extremely difficult to infer from them the intended behaviour of contractual agents. Although sanctions can help reduce the uncertainty that agents won't comply with the contracts, they may still decide not to fulfill parts or the entire contract and endure the associated sanctions. This implies that an agent cannot make the blind assumption that every party in the contract will behave as agreed initially.

Contractual agents based their decisions on their beliefs. These beliefs are influenced by the behaviour of the other agents and by the data coming from various enterprise systems. For instance, let's consider the following contract:

ns1:$O_{by,se}(pay(by, se, price, currency) < date(payment\_date))$

ns2:$fulfilled(ns1) \rightarrow O_{se,by}(deliver(se, by, product, quantity) < date(payment\_date + 30))$

If the buyer does not pay, the seller will never deliver as the obligation to deliver will never arise. However, if the buyer pays, the seller will be under the obligation to deliver and will have to decide whether to deliver or not. Given this analysis, one could think that the seller needs to wait for the buyer to fulfil or not his obligation before taking a decision on wether to pay or not. Adopting such strategy would result in a very safe but very poor contractual relationship management. On the other hand, if the seller knows that the buyer intends to pay then he might decide to execute some of the early stages of the delivery process (packing, custom, etc.) prior to the fulfilment of the payment obligation hence reducing the delivery time by a fair amount. We refer to this ability as *dynamic forecasting of partners behaviour*.

To achieve synchronization and forecasting, contractual agents base their communication on the Contract Fulfilment Protocol (CFP). CFP is a collaborative protocol based on the lifecycle of the deontic operators and is layered on top of the speech act theory (Searl 1969). As a detailed account of the CFP protocol is out of the scope of this paper, a set of messages illustrating how the CFP is used between contractual agents is presented below. Lets consider the contract described earlier on, the following CFP messages could be exchange in that context.

1. inform(buyer,seller,accept_norm(buyer,ns3))
2. request(buyer,seller,acknowledge_norm_executed(buyer,ns3))
3. inform(seller,buyer,acknowledge_norm_executed(buyer,ns3))

In the first message, the buyer informs the seller that he has accepted to execute the norm. This first message is typically used by the seller to infer that the buyers intention is to execute the action specified in the norm. This is an example of behaviour forecasting. In the second message, the buyer requests the seller to acknowledge that the norm ns1 has been executed, that is to say that the action specified in the norm has be carried out in accordance with the contract. The message is used to synchronize the views of the buyer and the seller with regard to ns1. The third message is an example of a possible message sent back to the buyer from the seller acknowledging the norm execution. As a result of that third message, both views are synchronized and both agent share the fact that ns1 is fulfilled.

Contractual Agents

Contrary to cooperative agents((Cohen & Lesvesque 1990), (Grosz & Kraus 1996), (Jennings 1995), (Pollack 1992)), normative agents are social entities, most often self-motivated, that take decisions and act based on existing social laws. In that model, cooperation cannot be guaranteed. Norms are used to reduce the uncertainty and make the behaviour of other agents much more predictable ((Castelfranchi *et al.* 1999), (Conte, Castelfranchi, & Dignum 1998)). Two approaches to norms have so far been developed. The first one considers norms as constraints on behaviour. Boman (Boman 1999) shows how these constraints can be used in the agent's decision process to ensure that no actions violating the norms could be taken. The second approach is to consider norms as external (expectations, behaviour and prescription) and internal (mental) entities. This allows agents to collectively issue norms, reason, communicate and negotiate about them. Castelfranchi (Castelfranchi *et al.* 1999) introduces the concept of deliberative agents that have explicit knowledge about the norms and can make a choice whether to obey them or not. In the same line, Barbuceanu (Barbuceanu, Gray, & Mankovski 1998) describes a model in which social laws are presented as objective forces that provide the ultimate motivation for coordinated actions. In his model, not fulfilling an obligation is sanctioned by a loss of utility or paying a cost. This allows an agent to apply rational decision making when choosing what to do.

Contractual agents find their roots in the approach introduced by Castelfranchi (Castelfranchi *et al.* 1999). However, our agents exhibit a more acute understanding of the notion of norms and in particular sanctions through their formal representation in a logical framework. Furthermore, they are able to communicate about the state of their contractual commitments in a way that is tightly coupled with their formal representation.

Figure 3, shows the conceptual architecture of the contractual agents.

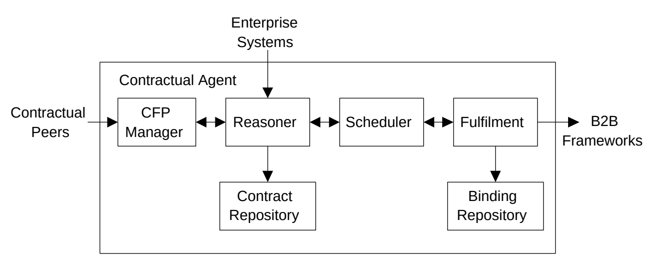

Figure 3: Conceptual Architecture of Contractual Agent

The CFP Manager is responsible for the communication with other agents. The Reasoner analyzes the contracts and its beliefs and select the normative statements to fulfil. The Scheduler manages the execution of the various normative statements based on their priorities. Through a binding mechanism (Sailer & Morciniec 2001), the fulfilment component creates a mapping between the actions specified in the contract and the processes in the B2B.

We present in Figure an high-level description of the algorithm used in the agent's reasoner.

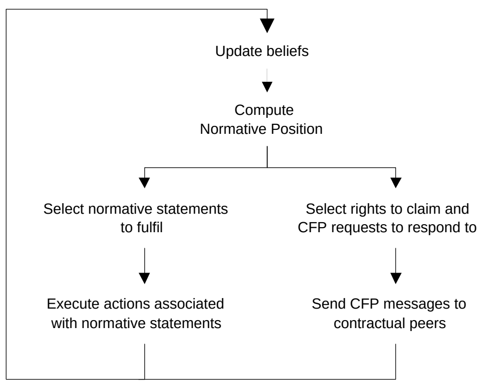

Figure 4: High-level reasoning process

The first step in this process is to update the knowledge base of the agent. The agent's belief are generated through the analysis of the CFP messages, the analysis of the messages received from the B2B framework and the data received from external enterprise systems. Once the agent's beliefs are updated, the agent computes its new normative position with regards to all its contracts. This results in a set of normative statements NS. From that set, the agent extracts O, the set of statements where he is the subject ($O \subseteq NS$) and decides whether to fulfilled them or not. This results in a subset E of O ($E \subseteq O$). This decision is based on assessing the utility for the agent of fulfilling a given normative statement. Sometimes, not fulfilling a norm and enduring the associated sanction might result in a higher utility than fulfilling the norm. For every normative statement in E, the agent executes the associated action.

From the same set NS, the agent extracts R, the set of all the rights (normative statements where the agent is beneficiary) granted to him. For every normative statement in R, the agent introspects its lifecycle state and could decides to send CFP messages either to ask the counterpart to start the fulfilment of the norm or to signal the counterpart that the norm has been violated. Finally, the agent might answer CFP requests received from counterparts.

Prototype

The Electronic Contract Framework prototype has been implemented using B2B state-of-the-art technologies. Contractual agents live in a J2EE [1] framework where each component of their architecture has been developed as an independent service with its own management interface. Internal communication between these components is done through JMS[2] and the contract repository is modelled as a set of session beans and container managed persistent entity beans. The CFP protocol has been specified as a set of XML messages over SOAP[3]/HTTP. The ECF is deployed and connected to HP Process Manager workflow engine as an instance of B2B framework. Public processes are defined using Ebxml BPSS[4] specification.

Conclusion

Automation of contractual relationships will enable flexible and dynamic trading across the internet. It will provide companies with an integrated approach to contract management insuring frictionless contract execution through real-time accuracy, compliance checking and contract lifecycle management.

In this paper, we have presented the concepts and the formalism used to model electronic contracts in our framework. We have then described how the dynamic nature of the normative statements are used by contractual agents to communicate about norms. Lastly, we have illustrated their use within the contractual agent's reasoning process.

This work shows that contractual agents connected to enterprise systems can play a major role in the automation of B2B contractual relationships.

References

Barbuceanu, M.; Gray, T.; and Mankovski, S. 1998. Coordinating with obligations. In Sycara, K. P., and Wooldridge, M., eds., *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, 62–69. ACM Press.

Boman, M. 1999. Norms in artificial decision making. *Artificial Intelligence and Law* 7:17–35.

Castelfranchi, C.; Dignum, F.; Jonker, C.; and Treur, J. 1999. Deliberate normative agents: Principles and architecture. In *Proceedings of The Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99)*.

Cohen, P., and Lesvesque, H. 1990. Intention is choice with commitment. *Artificial Intelligence* 42:213–261.

[1] Java 2 Enterprise Edition, http://java.sun.com/j2ee

[2] Java Messaging Service, http://java.sun.com/products/jms/

[3] Simple Object Access Protocol,http://www.w3.org/TR/SOAP/

[4] Business Process Specification Schema v1.01,http://www.ebXML.org

Conte, R.; Castelfranchi, C.; and Dignum, F. 1998. Autonomous norm-acceptance. In *M. Singh (ed.) Intelligent Agents*.

Dellarocas, C. 2000. Contractual agent societies: Negotiated shared context and social control in open multi-agent systems. In *Workshop on Norms and Institutions in Multi-Agent Systems, 4th International Conference on Multi-Agent Systems (Agents-2000)*.

Grosz, B., and Kraus, S. 1996. Collaborative plans for complex group action. *Artificial Intelligence* 86:269–357.

Jennings, N. R. 1995. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence* 75:195–240.

Kollingbaum, M., and Norman, T. 2001. Supervised interactions for electronic markets. In *To appear in proceedings of UKMAS 2001 workshop*.

Meyer, J.-J. C. Winter 1988. A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic*.

Morciniec, M.; Salle, M.; and Monahan, B. 2001. Towards regulating electronic communities with contracts. 2nd Workshop on Norms and Institutions in MAS.

Pollack, M. E. 1992. The uses of plans. *Artificial Intelligence* 57:43–68.

Preist, C. 2001. Agent mediated electronic commerce research at hewlett packard labs, bristol. *Newsletter of the ACM SIGecom Exchanges* 2.2.

Sailer, M., and Morciniec, M. 2001. Monitoring and execution for contract compliance. Technical Report HPL-2001-261, Hewlett-Packard Research Laboratories.

Searl, J. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge, England: Cambridge University Press.

van der Torre, L. 1999. Violation contexts and deontic independence. In *context*, 361–374.

von Wright, G. 1950. Deontic logic. *Mind* 60:1–15.