Integrated contract management

Abdel Boulmakoul, Mathias Sallè
Trusted E-Services Laboratory
HP Laboratories Bristol
HPL-2002-183
July 8th , 2002*

E-mail: {abdel_boulmakoul@hp.com, mathias_salle@hp.com}

e-services, business-to-business, electronic contract, contract management

This paper focuses on business-to-business contractual relationship in the context of second-generation contract management solutions. In particular, it describes the contract framework developed at HP Labs Bristol, UK. The presented solution is based on the concept of electronic contract, which represents a machine interpretable model for contracts. The contract framework has been designed with the view to support the whole lifecycle of the contract. It consists of a set of components to be deployed in each enterprise to allow peer-to-peer message based interaction with contract participants. The deployment model can be easily changed to a portal deployment where the contract participants use a browser client to initiate the interactions on the portal.

* Internal Accession Date Only
Approved for External Publication

Integrated contract management

Abdel Boulmakoul
Mathias Sallé
e-Business Technology Department
HP Laboratories, Filton Road, Stoke Gifford, BS34 8QZ Bristol, UK
{abdel_boulmakoul@hp.com, mathias_salle@hp.com}

Keywords: e-services, business-to-business, electronic contract, contract management

Abstract:
This paper focuses on business-to-business contractual relationship in the context of second-generation contract management solutions. In particular, it describes the contract framework developed at HPLabs Bristol, UK. The presented solution is based on the concept of electronic contract, which represents a machine interpretable model for contracts.
The contract framework has been designed with the view to support the whole lifecycle of the contract. It consists of a set of components to be deployed in each enterprise to allow peer-to-peer message based interaction with contract participants. The deployment model can be easily changed to a portal deployment where the contract participants use a browser client to initiate the interactions on the portal.

I. Introduction

Recently there has been renewed interest in modeling of business contracts in the academic computer science community [1,2] as well as in the industry [3]. This is motivated by the fact that enterprises increasingly use the Internet for communication with their partners and would like to leverage this technology in order to gain efficiency in contracting processes.
Contracts are important in the context of loosely coupled structures [Marshall, 1999] like supply chains that involve independent entities. Because there is no central authority that coordinates activities of entities making up a supply chain, each entity is responsible to arrange a contract with their partner defining the collaboration in which they will engage.
In real life, contracts define rights and obligations of parties as well as conditions under which they arise and become discharged. The rights and obligations concern either states of the affairs or actions that should be carried out. Often contracts also specify secondary (reparation) obligations that come into force when a party does not carry out an obligation. The essence of contracts is the definition of commitment states that is imposed on contracting parties. These states come into force and become discharged as a result of actions that the parties carry out or as a result of an occurrence of an external event such as expiration of a deadline.
During the contract fulfillment, parties collaborate by exchanging information and carrying out actions agreed upon them. They do so because of the contract that imposes commitment states on them. In normal circumstances parties aim to fulfill their responsibilities but it is perfectly admissible that a party will refuse to carry out an agreed action or refuse to maintain agreed state of affairs activating a secondary obligation. This situation typically occurs if an unforeseen event takes place (e.g., import restrictions) forcing one to de-commit from the obligation.
So far, contracts have usually been treated as merely text documents. However, combining well-structured information such as agreed terms and conditions into a single document, they promise suitability for driving an electronically automated

system. Thus, the resulting e-contract can provide a high degree of consistency and ease of use.

II. E-Contract Lifecycle

Conceptually, the lifecycle can be split into the three stages of contract drafting, formation, and execution [4]. Figure 1 illustrates the contract lifecycle.

- Contract drafting phase: Given the contract template model, the drafter role constructs an instance of the template. In this phase the contractual roles, abstract business interactions and contractual situations are specified. The template typically has a number of free variables that are agreed upon in the next phase.

- Contract formation phase: Participants assume contract roles and negotiate the details of their responsibilities. The negotiable variables of the contract (deadlines, order of actions) become fixed and concrete business interactions are bound to the abstract ones defined in the template. The relationships between contract parties are created and captured in contract statements. The statements contain policy expressions that imply obligations and rights of parties.

- Contract fulfillment phase: Actual delivery of contract consideration takes place. Typically, this phase constitutes service or goods delivery, invoicing, bill calculation, presentment and payment. The interactions between the parties are monitored for their compliance to the terms agreed on in the contract.

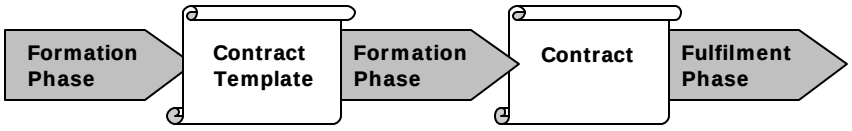

Figure 1: Contract lifecycle

III. E-Contract

Electronic contracts (e-contracts) formally specify the behavior 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 fulfill one or several of their contractual commitments.

An e-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, e.g. HP's a Buyer.
- 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 behavior of the various roles defined in the informative section. Normative statements are based on the operators of deontic logic introduced in [5] A formal representation of a normative statement is given below:

ns: f -> ?s,b(a<?)

Where:

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

Normative statement are read as follows: ''if f holds then s is {obliged, permitted, prohibited} {to/by} b to achieve a {before/until} ? holds true''.

Let's consider the following contract:

ns1: The seller must deliver the requested goods to the buyer within 45 days of the receipt of a purchase order.
ns2: The buyer must pay the seller within a month after receipt of the ordered goods.
ns3: If the seller fails to deliver the goods according to the terms specified in ns1, the seller will pay a penalty of 5% of the total price of the ordered good with 10 days of the agreed delivery date.
ns4: If the buyer fails to pay the seller as specified in the clause ns2, the buyer shall pay a penalty fee of 10% of the total price of the ordered good within 10 days of the agreed payment date and shall pay the seller the total price of the good with 30 days of the initial payment date.

We now present how this contract can modeled using normative statements:

ns1: receive_po(se,by,product,quantity,po_rcv_date) ->
$O_{se,by}$(deliver(s,b,product,quantity)<date(po_rcv_date+45))

ns2: fulfilled(ns1) ->
$O_{by,se}$(pay(b,s,product_price*quantity)<date(delivery_date + 30)

ns3: not_fulfilled(ns1) ->
$O_{by,se}$(paypenalty(by,se,price,5%)<date(delivery_date+10))

ns3: not_fulfilled(ns2) ->
$O_{by,se}$(paypenalty(by,se,price,10%)<date(payment_date+10)) &
$O_{by,se}$(pay(b,s,product_price*quantity)<date(payment_date+30)

IV. Contract Framework

The contract framework has been designed to support automation of contractual relationships. It supports two different deployment models to accommodate with customer requirements: *distributed* or *portal*.

The distributed approach allows two enterprises to connect their contractual framework through a message-based communication bus. Figure 2 presents the distributed deployment model. In such a mode, each contractual framework is respectively connected to the B2Bi systems of the company. Upon deployment of a contract in the frameworks, the execution of the various commitments of the parties will be automatically triggered in the execution frameworks.

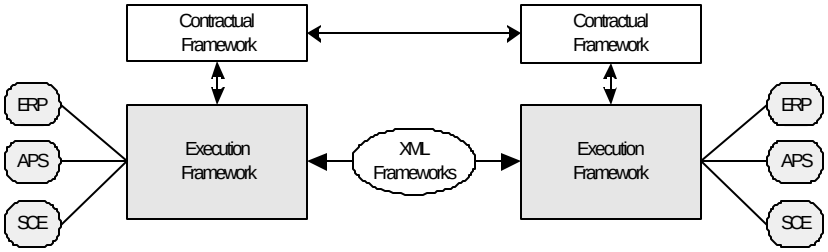

Figure 2: Distributed deployment model

On the other hand, the framework can be deployed as a portal. In this mode, only the company hosting the framework can benefit from the integrated automation of the framework.

The framework is based on the J2EE [6] architecture. All components are EJBs [7] or JMX [8] beans managed by the Jboss 2.2.2 [9] application server. The extra-enterprise messaging is over SOAP [10] and HTTP. The presentation layer is based on JSP [11] and servlet technology using Tomcat 3.2.2 [12] as a servlet container. This allows browser-based access to the contract framework regardless of the final deployment model.

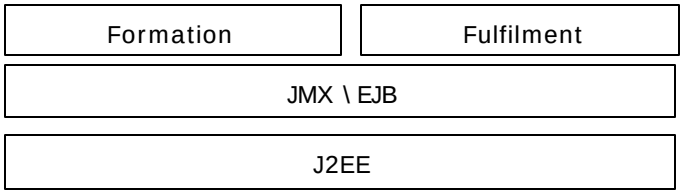

Figure 3: Technology stack

The framework has functionalities that fall into two categories: contract formation and contract fulfilment. These functionalities and their current state are discussed below.

IV.1 Contract Formation

Contract templates are used as initial inputs to the contract formation. Templates are unbound or partially bound contracts and traders negotiate over their unbound parameters. During this negotiation process, the traders exchange, counteroffer, reject or accept proposal. The Contract Negotiation Protocol (CNP) is specified to allow traders to modify proposal by not only changing the value of some open parameters e.g. price, delivery date, but by removing or adding new clauses. Once the template is fully bound and that the negotiating parties are satisfied by the outcome of the negotiation, the template is then signed.

CNP has been designed on top of SOAP/HTTP. The exchanged templates are serialized has XML documents exchanged as part of the SOAP messages. To maintain a consistent audit trail of the negotiations, each proposal is stored in a persistent store. Figure 4 gives a high level overview of the architecture of the formation module.

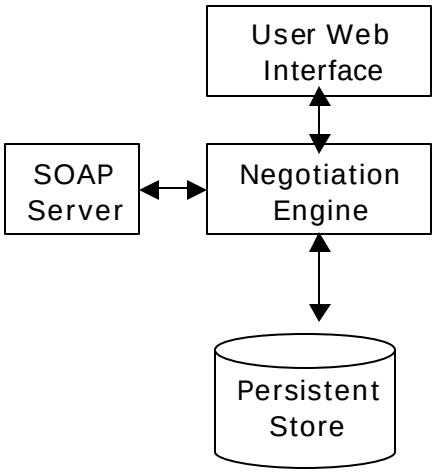

Figure 4: contract formation architecture

IV.2 Contract Fulfilment

IV.2.1 Contract Fulfilment Protocol (CFP)

By nature, contractual relationships are highly distributed. This introduces differences in the view that each contractual agent has on the contractual commitments.

To synchronize their views, 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 [13]. 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.

Let's 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)

Figure 5 presents the conceptual architecture of the Electronic Contract Framework. It is composed of four main components each assuming specific responsibilities. When a contract is deployed in the framework, it is added to the contract repository and is immediately monitored by the reasoning engine.

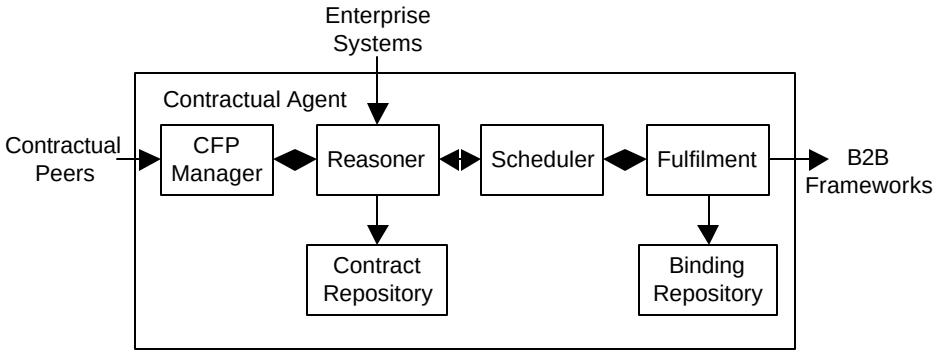

Figure 5: Conceptual Architecture

We now present in further details the four main components of the conceptual architecture.

IV.2.2 CFP Manager

The CFP manager is responsible for the communication with other contractual framework. It validates the syntactic and semantic correctness of the exchanged messages.

IV.2.3 Reasoner and Scheduler

The reasoner is the core component in the framework; it analyses the contracts and its beliefs and select the normative statements to fulfil.

The first step in the reasoning process is to update the knowledge base of the system. The reasoner's beliefs 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 beliefs are updated, the reasoner computes the new normative position with regards to all its contracts. This results in a set of normative statements NS. From that set, the reasoner 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 system 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. The reasoner forwards every normative statement in E to the scheduler for execution. From the same set NS, the reasoner extracts R, the set of all the rights (normative statements where the system is beneficiary) granted to him. For every normative statement in R, the reasoner introspects its lifecycle state and might 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.

IV.2.3 Fulfilment

The fulfilment component is the interface between the B2B framework and the reasoning component. It maintains the mappings between contract commitments, interaction schemes and execution instances. When a request is received to carry out a contractual commitment this component identifies the contract action that should be executed. For this action, an action executor is identified that handles the creation of an action instance and the monitoring of this instance for completion.
Collaborative actions in the contract are specified using the ebXML BPSS (Business Process Specification Schema) specification. A translator is then used to convert BPSS collaboration to a set of process templates. As a specific instance, our translator generates a set of HP Process Manager 4.0 process templates that implement the transactional interactions specified in BPSS [26, 27, 28].
A document event store is used to store events relating to the documents created, sent, and received as a result of action instances carried out in the execution subsystem. These events are then accessible from the reasoning engine as external knowledge used in the decision making process.

V. Conclusion

In this paper we have presente d a contractual framework that manages the whole contract lifecycle: drafting, formation and fulfilment. We also explored the possibility of expressing contracts electronically and we have given an abstract model for a contract statement that prescribes an action. We further showed how this abstract model of contract could be used to drive execution of contract and

the relationship of contract components with other enterprise components such as workflow manager or message manager.
In order to evolve and make use of our results, we are expanding the notion of electronic contracts to service level agreements [14,15].
Service Level Agreements represent non-functional properties of transactions, set of transactions or processes. For instance, "On average, per month, HP shall deliver VW the ordered goods within 10 days of the receipt of a purchase order." refers not to a specific transaction but to an average over a set of transactions that occurred during a month. SLAs can be expressed in our framework as constraints reflecting a state-of-affairs that should be brought about and possibly maintain. Thus, the previous SLA could be modelled as $O_{hp,vw}$ (average(delivery_date(x) - po_rcv_date (x)) < 10). Further work will develop an appropriate constraint language to define non-functional properties.

VI. Appendix:
Below we include short discussion on the law and security requirements with respect to electronic contracts. Interested readers are encouraged to follow indicated references for more detailed information.

VI.1 Legal Status of Electronic Contracts
The global pervasiveness of the Internet and the ability to use it as a medium for economic interactions enables international trading for partners who otherwise would never have met before. This revolutionary development leads to business scenarios and relationships that have not been previously foreseen.
The work that is described in the previous sections postulates that in the near future contracts will be drafted, formed and enforced on-line. This requires that established law and governments bodies evolve towards the vision of electronic society and pass appropriate legislation. The worldwide legislation effort is limited to that of United Nations Commission on International Trade Law (UNCITRAL) aimed at establishing a global legal framework for EDI [16], but most notable efforts so far can be observed in the European Union where the EU Directive on Electronic Commerce [17] has recently been passed. The directive lists a set of common rules for the conduct of electronic trade. It states that if the services provided by the service provider are lawful in a member state than the provision of these services in other member state is also lawful in other member state. This "country of origin" principle aims to protect the service providers and applies to B2B interactions. The rules laid out in the directive reinforce existing laws of Rome and Brussels Conventions that regulate which legislation is applicable (typically the law of the country most closely connected with the contract prevails). As many deployment models exist, determining the governing legislation domain can be complex [18] but this issue can be resolved within the European law framework.
The consumer contracts are an exception where the "country of destination" principle prevails. It states that customers will be able to sue under local consumer protection laws if dissatisfied with the service provided in a different country. This issue is further laid out in the directive [19] on the protection of consumers in the respect of distance contracts and specifies the rights for cancellation of the contract, deadlines and procedures for refunds in case of non-performance, arrangements for payment and delivery etc.
The Directive on Electronic Commerce with regards to electronic contracts also places certain information requirements that must clearly be given:

- An outline of different technical steps to follow to conclude contract;
- If the contract is to be filed or will be made accessible;
- Technical means for identifying and correcting errors prior to placing of the order;
- The languages in which contract can be concluded;
- Codes of conduct to which service provider subscribes.

A separate directive exists [20] that stipulates validity of electronic signatures in electronic contracts (although exceptions exist) and their admissibility in legal proceedings. Consequently the electronic contract cannot be dismissed purely on the basis that it is not paper based.
The implication of the on going legislative process on the requirements and guidelines for electronic contract system designers is not entirely clear. On going research exists that focuses on determining the impact of law on the design of the data structures and protocols [21, 22]. It is likely that in the future [23] there will be strong requirements on the modelling of contracts as the contract law for the virtual marketplaces matures.
Law relationship management is a new emerging area and although at the moment there are few start-up companies [24] globalisation of business interactions may generate a larger market for professional services in this area.

VI.2 Security requirements

Depending on the deployment of the contract framework and weather the communication takes place across the Internet or a virtual private network different security mechanisms will be required. Also one may consider the necessity to build in security mechanisms into various protocols [25, 2], present in throughout contract lifecycle. In this report we are not concerned with the security issues as these are likely to be known only once the business and risk model of the enterprise. Nevertheless, we list the security concerns for the consideration below:

- Authorization: The contract should be protected against improper, unauthorized access. Only people/agents who are signatories of the contract should be allowed to do so.
- Integrity: The contract should be protected against modifications as soon as at least one signature has been placed on the contract.
- Non-repudiation: The contract participants should not be able to deny their contractual commitments.
- Confidentiality and Privacy: The contract content should only be revealed to the signatories and/or trusted third parties that they have nominated.
- Authentication: the system must be able to identity contract participant.
- Valid signatures: The contract has to have valid signatures of the contract parties so as to have and expression of commitment to contractual obligations.
- Transport security: the communication channel between the contract participant should be secure.
- Secure Messaging: the business messages between the contract parties should be done in a secure manner.
- Secure storage: the contract document should be stored in a storage that will guarantee that it can be retrieved and processed even after a long time has elapsed.

VII. References

[1] Milosevic Z. et al, 1995, “An Architecture for Supporting Business Contracts in Open Distributed Systems”, Proceedings of the Second International Workshop on Services in Distributed and Networked Environments (SDNE95), pp.60-67, Whistler, Canada.

[2] Milosevic Z., Arnold D., O’Connor L., 1996, ”Inter-enterprise Contract Architecture for Open Distributed Systems: Security Requirements”, WET ICE 1996, Workshop on Enterprise Security, Stanford USA.

[3] Ibbotson J., Sachs M., 1999, “Electronic Trading Partner Agreement for E-Commerce”, IBM Corporation.

[4] Boulmakoul A., Bartolini C., Morciniec M., 2000, “Electronic Contract Specification”, HP Labs Bristol

[5] G. von Wright. Deontic logic. Mind, 60:1–15, 1950.

[6] http://java.sun.com/j2ee/

[7] http://java.sun.com/products/ejb/

[8] http://java.sun.com/products/JavaManagement/

[9] www.jboss.org

[10] http://www.w3.org/TR/SOAP/

[11] http://java.sun.com/products/jsp/

[12] http://jakarta.apache.org/tomcat/tomcat-3.2-doc/index.html

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

[14] Alexander Keller, Gautam Kar, Heiko Ludwig, Asit Dan, Joseph L. Hellerstein: “Managing Dynamic Services: A contract based approach to a conceptual Architecture”

[15] Sahai, Akhil; Durante, Anna; Machiraju, Vijay : Towards Automated SLA Management for Web Services

[16], Heinrich G., 1995, “Harmonized global Interchange? UNCITRAL’s draft model law for electronic data interchange”, Journal of Current Legal Issues 3, http://webjcli.ncl.ac.uk/

[17] ”Directive 2000//EC of the European Parliament and of the Council of 4 May on certain legal aspects of Information Society services, in particular electronic commerce in the Internal Market (Directive on Electronic Commerce)“, http://europa.eu.int/comm/internal_market/en/media/eleccomm/index.htm

[18] Susman A. M., 2000, “International Electronic Trading: Some Legal Issues” in Computers and Law 10-6, pp.29-30.

[19] “Directive 97/7/EC of the European Parliament and of the Council of 20 May 1997 on the Protection of Consumers in respect of Distance Contracts”, OJ 1997 L144/19.

[20] “Directive 1999/93/EC of the European Parliament and of the Council of 13 December on a Community framework for electronic signatures”, OJ 2000 L13/12.

[21] “Can Computers Make Contracts?”, Harvard Journal of Law & Technology, 9, No. 1, http://jolt.law.harvard.edu/archives.html

[22], Gisler M., Stanoevska-Slabeva K., Greunz M., 2000 “Legal Aspects of Electronic Contracts”, ISDO 2000, Stockholm, 5-6 June 2000.

[23], Favier J., 2000, “eMarketplaces Face the Law”, the Forrester Report, October 2000, http://www.forrester.com

[24] EGlobal Law, October 2000, http://www.egloballaw.com/

[25] Gisler et al., 1999, “Secure Electronic Contracts”, SPP ICS Project, working paper no 1, Requirements on Electronic Contracts, MCM Institute, Switzerland.

[26] Electronic Business XML initiative, http://www.ebXML.org/

[27] RosettaNet Consortium, http://www.rosettanet.org

[28] http://www.ebxml.org/specs/index.htm