CAN AGENTS CLOSE CONTRACTS?

Frances Brazier[1], Anja Oskamp[2], Maurice Schellekens[3] and Niek Wijngaards[1]

[1] Intelligent Interactive Distributed Systems, Faculty of Sciences
Vrije Universiteit Amsterdam, de Boelelaan 1081a, 1081 HV, Amsterdam, The Netherlands
Email: {frances,kubbe,niek}@cs.vu.nl
Phone: +31 - 20 - 444 7737, 7756; Fax: +31 - 20 - 444 7653

[2] Computer and Law Institute, Faculty of Law
Vrije Universiteit Amsterdam, de Boelelaan 1105, 1081 HV, Amsterdam, The Netherlands
Email: a.oskamp@rechten.vu.nl
Phone: +31 - 20 - 444 6215; Fax: +31 - 20 - 444 6230

[3] Center for Law, Public Administration and Informatization, Faculty of Law
Tilburg University, P.O. Box 90153, 5000 LE, Tilburg, The Netherlands
Email: M.H.M.Schellekens@uvt.nl
Phone: +31 - 13 - 466 - 8044; Fax: +31 - 13 - 466 8149

Abstract. One of the tasks of software agents on the Internet will be to close contracts on behalf of their owner. The closing of a contract is subject to liabilities. In this, there is no distinction between the real world and the virtual world. Like in the real world, within the virtual world a distinction has to be made between the precontractual phase and the contractual phase. The liabilities in these phases are different. Therefore negotiation protocols for agents need to distinguish between these phases and different responsibilities may exist in the different phases. This paper presents some of the issues involved, viewed from a technical and legal perspective.

1. Introduction

The Internet and its advent of E-commerce applications involving software agents (He and Leung, 2002) gives rise to the issue whether software agents can legally close contracts. Technically, software agents are autonomous and pro-active, can autonomously (Castelfranchi, 2001) roam the Internet, perform transactions, and gather information. It can be questioned though whether agents are allowed to act autonomously from a legal point of view, especially when performing legal acts, like closing a contract. This paper presents a view on contract negotiations between agents combining the research areas Law and Computer Science, addressing Dutch law and relevant EU legislation.

Negotiations between parties are a means to agree upon issues that may result in the closure of a contract. However, not all negotiations lead to a contract; sometimes negotiations fail. In such a case, parties go their separate ways without further obligations. Sometimes, however, the negotiations are not fully noncommittal. Costs have been incurred or expectations have been raised that may have led to, for instance, breaking off negotiations with other parties. In legal terms, this is the domain of precontractual liability; explicitly present in civil-law, but not in common law. When negotiating with other software agents, software agents may also encounter the problems referring to the distinction between the various phases and obligations and liabilities connected to these phases. This problem is dealt with in the first part of this paper.

The actual contracting by software agents raises other concerns: what is actually needed to agree on a contract? How can software agents technically comply with these requirements? Are there good practices to which a software agent can adhere to make the process of contracting run smoothly? These questions are the subject of the second part of this paper. The paper concludes with a discussion.

2. Pre-contractual liability

2.1 Legal analysis

Based on Dutch case law, it is possible to distinguish between three stages within the process of negotiations (Hartkampand Asser, 2001). The distinction between the stages is made on the basis of good faith.

Stage 1: Parties are free to break off negotiations without any obligation to compensate the other party;

Stage 2: According to the criterion of good faith a party is still free to break off the negotiations. However, there is the condition that he compensates the other party for (all or some of) the expenses made;

Stage 3: According to the criterion of good faith a party is no longer free to break off the negotiations. If that party nevertheless breaks off the negotiations, he is obliged to compensate the other party for expenses and - in some cases - for the profits the other party would have made, if the contract had been closed.

In the process of negotiation, these stages are not formally demarcated; the stages are made explicit and discussed after a dispute has arisen in the pre-contractual sphere. To give the users of software agents some support with respect to issues of precontractual liability, we propose to accommodate the different stages in a rudimentary form in the protocols used for agent-negotiation. A ‘rudimentary’ form is chosen because the open texturedness of the concept of ‘good faith’ is difficult to catch in a computer-understandable conceptualisation and perhaps impossible.

When distinguishing between the various phases the concept of ‘good faith’ is not taken as a starting point, but a number of events that are indicative for the different stages: a pragmatic approach. Software agents may recognise a number of events that are indicative for later problems and may support their respective users in taking appropriate actions. Examples of relevant events are:

1. event: a party is about to incur costs for the preparation of the contract;
   relevance: who bears the costs in case the contract does not materialise?
2. event: a party breaks off parallel negotiations;
   relevance: the party becomes more dependent upon successful conclusion of the negotiations.

Negotiation protocols should be based on these events. If costs are about to be incurred, the parties’ software agents must agree about who bears the costs. If an agent is about to break off parallel negotiations the agent must try and agree with the agent with which he is to proceed that henceforward the breaking off of negotiations means that the other party must be compensated for possible losses. The incomplete character of the recognition of the stages, means that the user of the software agent must be adequately warned: the software agent can recognise some pitfalls of pre-contractual liability, but not all. The mere recognition of the existence of these events and the phases may be important for the assignment of liability. The details of this recognition under the various legal systems is still a topic of further research.

2.2 Technical issues for pre-contractual phase

The importance of well-defined protocols of negotiation is evident. At least, which legal system holds, which general conditions apply and which obligations hold at which moment / during which phase, needs to be communicated/advertised in non-ambiguous terms. In addition, protocols may define specifics about message delivery (message is sent versus message is received semantics). It is unclear if this is possible, since it would mean interpreting certain concepts and trying to put them in non-disputable formats, covering various legal systems. It is a topic of further research to discover whether this is possible: if not, to what extent this would require general directives concerning the use of software agents, including adaptation of the various legal systems.

It is not common for E-commerce protocols to distinguish between a pre-contractual phase and a contract closure phase; they usually focus on the latter phase (e.g., Sherif, 2000) in which non-repudiation and authenticity play an important role. Note that this may be due to the legal system assumed. Below, a number of protocols is briefly described:

- The Secure Electronic Transaction Protocol, (Lu and Smolka, 1999), focuses on secure payment card transactions (e.g., by Visa and MasterCard).
- The Internet Key Protocol (Bellare, Garay, Hauser, Herzberg, Krawczyk, Steiner, Tsudik, and Waidner, 1995) consists of a number of secure electronic payment protocols, with varying degrees of non-repudiation and authentication.
- The fair non-repudiation protocol (Zhou and Gollmann, 1996) is more general and involves a trusted-third party to ensure fairness about message reception and integrity.
- The open trading protocol (Bichler, Segev and Zhao, 1998) is a broader protocol for interoperability of electronic purchases, including payment, invoices and delivery.
- The Needham-Schroeder public key protocol is also more general and establishes mutual authentication between initiators and responders.

A view on the general and specific protocols described above is that their basic line may be used for assuring 'statements' between parties, and form a basis for protocols distinguishing the two phases. More research is clearly required on this topic, e.g. in the context of contract management systems, in which a pre-contractual phase is conceptually distinguished, but not operationally included (Boulmakoul and Sallé, 2002).

Protocols employ languages and ontologies such as the WSDL, OWL (successor of DAML-S and OIL; Patel-Schneider, Horrocks and van Harmelen, 2002), or RDF (W3c.org). Most agent communication languages (ACLs) (e.g., Fipa ACL, 2002) upon which most communication between agents is based, currently do not include primitives for these phases, nor do they include properties describing the phases of negotiation/ contract assessment, although they do provide negotiation primitives. This is an area that needs further attention within the web and agent community.

There are logics in which, e.g., beliefs, intentions, commitments, good faith are expressed (e.g., Rao and Georgeff, 1993). Verification of an agent's status on the basis of such logics in which an agent's reasoning is expressed may be possible in a number of cases. The question is, however, whether courts of law are willing to accept proofs expressing an agent's intent/beliefs as legal evidence. The implementation of such systems is, in most cases, not fully tractable nor verifiable, possibly making evidence unreliable.

Another point of consideration is whether strategies can and/or should be conveyed. To the extent that strategies are known (see literature on game theory), this may be possible, but most likely not desirable. Time constraints in negotiation may, in some cases, be a point of concern. Technical guarantees for all real-time constraints are not currently feasible. If mechanisms for fault-tolerant agent systems such as those being developed by (Marin, Sens, Briot and Guessoum, 2001) are included in new agent platforms, a certain level of certainty may be provided for specific classes of constraints.

Agents and agent platforms (which support agents) may be obliged to trace their actions. Traceability involves logging information about agent actions, a process which leads to large amount of tracing data, possibly distributed among multiple agent platforms in different legal domains. Determining the granularity of the actions logged, the reliability of tracing data, and storing and processing such tracing data may not be easily accomplished and requires more research.

3. Contract formation

3.1 Legal analysis

What implications do contract law and legal practice have for the use of software agents for contract formation? With respect to the actual contracting the requirements that a software agent must meet can be divided into three categories:

1. basic requirements for the contract to be formed in a legally valid way,
2. compulsory rules that must be adhered to (e.g. to protect weaker parties) and
3. good practices (e.g. the collection of evidence).

1. Basic requirements

Contracts come about by statements. The user uses the software agent as a means to make a statement (Weitzenböck, 2001; Esch. 1999). In order for the software agent to function correctly as a means for making statements the following requirements must be met:

- The user of the software agent must have control over the (goals of the) software agent; e.g. he must be able to switch the software agent on or off, he can instruct the software agent, etc.
- The software agent must be transparent to the user: the user must be able to foresee what the software agent is going to do/declare. Without transparency, control by the user is illusory.
- The actions of the software agent must not be corrupted; the software agent must be reliable and it must have sufficient security built into it.
- The sending and receiving software agents must be 'attuned to each other' in order to prevent miscommunication.

A different basic requirement concerns the legal capacity of the user of a software agent to close a contract. A minor lacks, e.g., the capacity to close a contract. This issue is not dealt with here, but it is an issue that is a major discussion topic between lawyers dealing with software agents. The final conclusion has, as yet, not been reached.

2. Compulsory rules

The compulsory rules as found in the Directive 2001/31/EC and the Directive 97/7/EC address information duties (Lodder and Voulon, 2002). From a technical perspective it is not so much the content of those information requirements that is relevant; these are for lawyers to fill in once technical format has been found. It is rather the form and the time at which the information is to be supplied to the other contracting party that determine the technical decisions to be taken.

For example, consider the form requirement: in certain circumstances, art. 5 Directive 97/7/EC requires a supplier to send to a consumer a 'written confirmation or confirmation in another durable medium available and accessible to him'. If the consumer uses a mobile software agent for communication with the supplier, there may be a problem: can storage in a software agent considered to be durable? Are there any requirements that may make it durable if met. Is the information stored in an agent available to the consumer if the agent resides at a remote location? The answer to these questions is not clear, but an answer in the negative is not unlikely. The problem is not made easier by the fact that the Directive burdens the supplier with the duty to provide the information in the required form, but the consumer cannot be ordered not to use a mobile agent. Technically a solution may be found in that an ACL provides for a way to indicate that information sent to the mobile agent of the recipient must immediately be forwarded to the user of the software agent and stored on a durable medium, such as the hard disk of the user. The supplier can then configure his information in such a way that the receiving software agent understands what is expected (forward to user and store on his hard disk).

3. Good practices

Evidence must be collected during contract formation in order to prove the existence and contents of the contract at a later date. More specifically, it is desirable to have evidence about the following facts:

- The identity of each party to the contract (see below).
- The authentication by each party. Authentication means that a party has adopted the contents of the contract as his own. Typically, a signature (or an electronic signature) is used for authentication purposes. With respect to the proof of the authentication, several subproofs can be discerned. In the first place, the fact of the adoption itself must be provable. In the second place, it must be clear who adopted the contents of the contract. In the third place, the adoption by a party must provably concern the contents of the contract. The latter may be a problem if several versions of a contract exist or, in case of man-machine interaction, if the contract displayed on the screen differs from the contract that is electronically signed.
- The 'authority' of the software agent to make a statement. Can it be proven that the agent did or did not act under the influence of a bug or a fraud? Can it be proven that the instructions the user gave to the software agent, are the instructions on which the software agent has acted?
- The integrity of the contract must be preserved, between the moment of its coming into existence and the moment of evidence delivery.

Apart from evidence delivery, other aspects are in practice closely related to contracting.

- Confidentiality of communications. A party can have (one of) several reasons for keeping the contents of data confidential, such as competitive considerations, or prevention of fraud (e.g. in case of a credit card number)

3.2 Technical issues for contract closure phase

This section describes the agent specific technical measures that can be taken to close contracts. For most of the measures accepted techniques in, e.g. software engineering, human computer interaction and security, can be identified. This section only addresses agent specific measures.

Technical measures for the producer of the software agent:

- The software agent of the supplier supports structured negotiation that discerns the consecutive phases of negotiation and can recognise and speak different protocols.

Technical measures for the producer of the software agent and the supplier of goods:

- Software agents need to be able to structure negotiation and contract finalisation using a standard/agreed ontology of which the syntax and semantics is known.

- Agents need an Agent Communication Language (ACL)1 to interact and provide agents with the means to exchange information and knowledge. Various ACL's exist (e.g., FIPA, 2002). For the development and use of agents it is important to know the characteristics of these languages. Standard ontologies are needed with which contract negotiation and finalisation can be structured.

The producer of the software agent:

- The software agent of the supplier supports structured negotiation that discerns the consecutive phases of negotiation.
- There are three types of information exchange within the context of the closure of a contract: the exchange of information with the other party, the exchange of information with marketplace, the exchange of information with parties that need to know information in order to fulfil their part in the collective that constitutes the contracting party.

The consumer:

- It is advisable that the consumer makes public (or at least communicates to partners in negotiation) for what contracts the software agent is authorised, and to what extent.

The producer of the agent:

- It must be clear to the user of the software agent what the authorisations and the instructions of the software agent are.
- The software agent (of the consumer) must have means to communicate to another (the supplier) its authorisations and instructions.

The agent platform:

- Support for authorisation should be provided by agent platforms. Different techniques may be used. Certificates is one option, encrypted messages using a PKI another.
- Safeguarding of agents and their associated (contractual) data.

4. Concluding remarks

Contracting plays a role in many possible applications of software agents. E-commerce is an obvious example, but by far not the only one. This paper focuses mainly on applications within e-commerce, but many of the observations also hold for other contract forms.

What has become obvious is that closing a contract with the help, or even by means of a software agent is not easy from a legal point of view. The legal rules are all directed towards human beings or corporations. Using software agents means introducing a new and different legal situation. Not all legal rules can be applied to that situation. But also software agents have to be constructed in such a way that they meet the legal requirements for closing of contracts.

[1] The origin of ACL's can be traced back to the Knowledge Sharing Effort (KSE) that was initiated by DARPA in 1990 [Labrou, Y., Finin, T., Peng, Y, (1999)]. In KSE researchers from both academia and industry co-operated. The goal of the KSE was to develop techniques, methodologies and software tools for knowledge sharing, reuse for design, implementation or execution.

This paper has named some of the questions about the legal requirements for legally valid closure of contracts by software agents. It has also addressed some of the technical possibilities to meet with these requirements. It is important to note that this paper only addresses and poses the questions and does not yet give full solutions to the problems indicated. For this a lot more research is necessary as often stated in this paper. It is, however, clear is that software agents also need to distinguish between the precontractual and contractual phase, since the obligations, liabilities and according requirements, both technical and organisational, differ in these phases. It is unclear whether a software agents will ever be able to distinguish between these phases on its own. The border between these phases is rather diffuse and dependent on the interpretation of concepts like good faith. The problem may be partly solved by agreements and standards incorporated in generally agreed upon protocols for agents. It remains to be seen whether such solutions can also be used to handle the difference between legal systems. We would strongly recommend to research the possibilities to reach agreements on these topics and incorporate those results in negotiation protocols.

Other topics that need to be researched more closely are the necessity for traceability of an agent's actions referring to evidence in negotiations. How secure and durable should this information be? To keep all information may lead to an information overload. What is legally necessary, minimum and maximum? Can standards for this be defined? Can this information be secured, so it cannot be interfered with in a later phase, thus validating its authenticity to make it valid legal proof?

These and many more questions will actually rise when agents roam the Internet and are able to close contracts. Now it is the time to research and explore these topics to formulate the measures needed in the future.

Acknowledgements

The ALIAS project is supported by NLnet Foundation, `http://www.nlnet.nl/`. The ALIAS project is a multi-disciplinary project specifically aimed at exploring the legal status of agents and the implications of their use. The authors wish to acknowledge the contributions made by the participants of the ALIAS project: Martin Apistola, Onno Kubbe, Corien Prins, Erik Schreuders and Marten Voulon; `http://www.iids.org/alias/`.

References

Bellare, M., Garay, J.A., Hauser, R., Herzberg, A., Krawczyk, H., Steiner, M., Tsudik, G., and Waidner, M. (1995), *iKP - A Family of Secure Electronic Payment Protocols*, IBM

Bichler, M., Segev, A., and Zhao, J.L. (1998), Component-based E-commerce: Assessment of Current Practices and Future Directions, *SIGMOD*, **27**(4): 7-14.

Boulmakoul, A and Sallé, M. (2002), Integrated Contract Management, *Proceedings of the 9th workshop of HP Openview University Association*, paper 4.1.

Brazier, F.M.T., Kubbe, O., Oskamp, A., and Wijngaards, N.J.E. (2002), Are Law-Abiding Agents Realistic? *in* Sartor, G. and Cevenini, C. (Eds.), *Proceedings of the workshop on the Law of Electronic Agents (LEA02)*, pp. 151-155.

Castelfranchi, C. (2001), Again on Agents? Autonomy: A Homage to Alan Turing, *in* C. Castelfranchi and Y. Lespérance (eds), *Intelligent Agents VII*, Lecture Notes in Artificial Intelligence, **1986**, Springer-Verlag, Berlin Heidelberg, pp. 339-342.

Cavanillas, S. and Nadal, A.M. (1999), *Research paper on contract law*, Electronic Commerce Legal Issues Platform (ECLIP) report, Deliverable 2.1.7bis.

Esch, van R.E. (1999), Electronic Data Interchange (EDI) en het vermogensrecht (diss. Nijmegen), Deventer: W.E.J. Tjeenk Willink.

Fipa ACL (2002), *Fipa ACL Message Structure Specification*, SC00061G, Standard, 2002/12/03, and *Fipa Ontology Service Specification*, XC00086D, Experimental, 2001/08/10.

Hartkamp, A.S. and Asser, C. (2001), *Asser's Handleiding tot de beoefening van het Nederlands burgerlijk recht. 4. Verbintenissenrecht. Deel II. Algemene leer der overeenkomsten*, Deventer: W.E.J. Tjeenk Willink, pp. 145-159.

He, M. and Leung, H-F. (2002), Agents in E-Commerce: State of the Art, *Knowledge and Information Systems*, **4**, pp. 257-282.

Labrou, Y., Finin, T. and Peng, Y. (1999), The current landscape of Agent Communication Languages, *IEEE Intelligent Systems*, **14**(2): 45-52.

Lodder A.R., and Voulon M.B. (2002), Intelligent agents and the information requirements of the Directives on distance selling and E-commerce, *The International Review of Law, Computers & Technology*, **3**, pp. 277-288.

Lu, S. and Smolka, S.A. (1999), Model Checking the Secure Electronic Transaction Protocol, *in Proceedings of Seventh International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'99)*, ACM Press, pp. 358-365.

Marin, O., Sens, P., Briot, J-P. and Guessoum, Z. (2001), Towards Adaptive Fault-Tolerance For Distributed Multi-Agent Systems, *in Proceedings of ERSADS'2001*, Bertinoro, Italy, pp. 195-201.

Patel-Schneider, P., Horrocks, I., and Harmelen, F. van (2002), Reviewing the Design of DAML+OIL: An Ontology Language for the Semantic Web, *in* R. Dechter, M. Kearns and R. Sutton (eds) *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pp. 792-797.

Rao, A. S. and Georgeff, M. P. (1993), A model-theoretic approach to the verification of situated reasoning systems, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)* , Chambéry, France, pp. 318-324.

Sherif, M.H. (2000), *Protocols for Secure Electronic Commerce*, CRC Press, Boca Raton, Florida.

W3C.org, for various topics, see http://www.w3c.org/

Weitzenböck. E.M. (2001), *Electronic agents and the formation of contracts*, ECLIP report 2001.

Zhou, J. and Gollmann, D. (1996), A fair non-repudiation protocol, *Proceedings of the 15th IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, pp. 55-61.