“Entity” and “Autonomy” – The Conclusion of Contracts by Software Agents in the Eyes of the Law

A Software Agent Definition-based Analysis

Tina Balke*

* University of Bayreuth
Chair of Information Systems Management (BWL VII), 95447 Bayreuth, Germany
tina.balke@uni-bayreuth.de

ABSTRACT. Faced with the ongoing evolution of software agents from mere passive tools to e-tailers acting autonomously for their human owners (principals), new legal challenges appear on the agenda. One of them is the question, how contracts concluded by software agents should be treated legally. Based on the characteristics of software agents identified from the definition of the term, this paper examines whether and how the conclusion of contracts by software agents is compatible with current legislation or poses problems that need to be addressed in the future. Thereby special focus will be on the autonomy property, as this one is highly disputed in legal terms.

RÉSUMÉ. Avec le développement continu d’agents logiciels, qui de simples programmes d’assistance passive ont évolué pour devenir des programmes électroniques agissant de manière autonome, des problèmes et défis juridiques nouveaux sont apparus. Dans ce contexte, la question se pose de savoir comment les contrats conclus par ces agents logiciels doivent être traités légalement. Se basant sur les caractéristiques des agents logiciels selon la définition du terme, cet article examine si et dans quelle mesure, les contrats conclus par ces agents sont compatibles avec la législation actuelle. Dans cet article, l’accent sera mis particulièrement sur la propriété d’autonomie des agents logiciels, ceux-ci étant extrêmement contestés en termes légaux.

KEYWORDS: Software Agents, Autonomy, Legal Aspects, Law of Agency

MOTS-CLEFS : Agents Logiciels, Autonomie, Aspect Juridiques, Contrats de Représentation

1. Trust as Requirement for Electronic Transactions

With the change of the agent paradigm from mere communication tools to autonomous “intelligent” software entities, the vision of these agents populating electronic markets, representing their users or owners (their principals) and conducting business on their behalf, has come into reach. However, thinking one step further, this vision includes the idea of software agents being able to initiate and conclude agreements by themselves without their principals being aware of these actions. Hence, agreements might no longer be generated through machines but by them, without the intervention or supervision of an individual (Weitzenböck, 2001). Software agents might therefore be employed as initiators and mediators of electronic transactions assisting human interaction through all stages of the transaction process. However, for this to happen on a large extent, the doubts of the potential users have to be allayed. That is why, as pointed out by Fasli (Fasli, 2007), in order for this vision to materialise, one fundamental issue that needs to be addressed is that of trust.

First, users need to be confident that their agents will do what they are intended to do. Second, they need to trust that their privacy is protected and that the security risks involved in entrusting agents to perform transactions on their behalf are minimised. Finally, users need to be assured that any legal issues relating to software agents trading electronically are covered to an extend as they are in traditional trading practices (Fasli, 2007).

Whereas the first two trust questions have been dealt with by many scientists, so far little results have been achieved when it comes to the legal issues concerning software agents. Some questions to be addressed in detail in the future are for example, as to whether contracts concluded by software agents are contractual binding on the parties, who is liable in case the software agent commits a mistake or causes any damage, or in how far the law of agency is applicable to software agents in this context (Weitzenböck, 2001).

Extending an article that has been presented at the 7th International Conference on Autonomous Agents and Multiagent Systems (Balke et al., 2008), this paper analyses the current legal situation concerning the third question posed above. Therefore, it starts by looking at the definition of software agents in section 2. Starting from the special characteristics that result from the definition, afterwards—in section 3—the consistency of these characteristics with the legal concepts for contract conclusions is analysed and discussed in detail. Furthermore, the existing legislation on the conclusion of contracts by software agents will be presented in section 4. After a short description of remaining legal issues in section 5, the paper finishes by focusing on remaining legal questions and by discussing proposed solutions (section 6).

1. All legal citations contained herein are based on current legal statutes in effect on July 11, 2012, the submission date of this paper. The authors cannot assume any liability for subsequent changes in pertinent laws which might have an impact on the analysis contained in this work.
2. Characteristics of Software Agents

When aiming at analysing software agents in the eyes of the law, first of all the term “(software) agent” itself has to be looked at and this is where the first problems arise.

Thus, so far a number of definition of the term “agent” has been provided (e.g. (Bradshaw, 1997, Jennings, 2000, Luck et al., 2003, Shoham, 1993)), however as Russel and Norvig (Russell et al., 1995) already in 1995 pointed out, although the term agent is widely used by many people working in closely related fields of research, so far no universally accepted definition could be agreed on. Thus, at the moment only a very basic description has been consented to, that is illustrated in figure 1:

\[\text{A [software] agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors}^2\]. (Russell et al., 1995, p. 31)

Besides this very abstract definition, a common set of auxiliary agent attributes that characterise software agents exists.

In the field of artificial intelligence (AI), a fairly comprehensive approach to define software agents using these common characteristics was proposed by Wooldridge and Jennings who use the common attributes of software agents to develop a “weak” and the “strong” notion of agency (Wooldridge et al., 1995).

In their weak notion they specify the following key terms as characterising software agents:

---

2. In case of a software agent these percepts and actions are encoded bit strings, as a software agent is a computer program.
autonomy the agent’s capacity to act without the intervention of its human principal or any other users and thereby having some level of control over its activities and internal states,

social ability the agent’s ability to communicate with other agents and humans through a shared agent communication language,

reactivity the agent’s ability to perceive an environment and respond in a timely fashion to changes that occur within it, and

pro-activity the agent’s ability to demonstrate goal-directed activity by taking initiative.

This list of attributes is expanded in Wooldridge’s and Jennings’ strong notion of agency in which they furthermore mention knowledge, belief, intention, obligation, mobility, veracity, benevolence and rationality as auxiliary characteristics for software agents and thereby attribute software agents all necessary characteristics to, at least virtually, support all stages of the contractual process.

Another comprehensive definition of the term agent was given by Ferber (Ferber, 1999, p. 9). He uses the abstract agent definition and combines it with the attributes mentioned by Wooldridge and Jennings as well as own attributes:

An agent is a physical or virtual entity
- which is capable of acting in an environment,
- which can communicate directly with other agents,
- which is driven by a set of tendencies (in form of individual objectives or of a satisfaction/survival function which it tries to optimise,
- which possesses resources of its own,
- which is capable of perceiving its environment (but to a limited extend),
- which has only a partial representation of its environment (and perhaps none at all),
- which possesses skills and can offer services,
- which may be able to reproduce itself,
- whose behaviour tends towards satisfying its objectives, taking account of the resources and skills available to it and depending on its perception, its representation and the communications it receives.

In case the “agent” in these definitions is a human person (entity), the question about the legal consequences of agents concluding electronic contracts is generally solved in legal terms. However, this situation changes rapidly and gets more difficult when referring the software agents instead (Allen et al., 1996). That is why, starting from the characteristics stated in the weak notion of agency and using the further characteristics for support, in the next section this paper will analyse and discuss the consistency of the characteristics and the legal concepts in detail. Thereby special focus will be on the “autonomy” characteristic, which is highly discussed by lawyers and AI researchers like-wise.
3. Software Agents in Legal Terms

3.1. Entity

The first characteristic of software agents that shall be looked at is the “entity” definition.

Looking at the current international legislation, both civil and common law define two essential elements for the validity of a contract: capacity and consent (Øren, 2001, Weitzenböck, 2001). Hence, to be legally valid and enforceable, a contract must have been entered by at least two parties that are capable of contracting in terms of the law and have expressed mutual statements of intention concerning the content of the contract. The capability of contracting means that in practice only natural persons (i.e. human beings) or legal persons 3 have the legal capacity to contract. Thereby, the law assumes the capacity to contract as the rule and defines exception-cases which lead to an incapacity. Hence, e.g. minors or aments are not considered to have the requisite capacity to enter a legally binding contract by the law (Weitzenböck, 2001).

Within these legal notions it is important to note that the absence of personality is not the same as incapacity, because incapacity always refers to a natural or legal person and therefore implies personality. In contrast, an entity or group from which the legislator has not conferred, or withdrawn personality, is not a person in the eyes of the law and therefore has no juridical life, meaning that no legal basis for attributing (or not) rights and obligations to it exists (Wettig et al., 2003).

Keeping this in mind, it becomes obvious that the use of software agents for the conclusion of contracts leads to considerable doctrinal discussions, such as: How can software agents that have not been attributed judicial personhood initiate contractually binding contracts for their principals and a third party? Or does the “entity” term imply a certain legal personhood and therefore does not conflict with the capability criterion?

An approach to deal with the above mentioned problems in case one assumes the “entity” term not automatically to imply legal personhood, is to consider software agents as mere passive communication tools and anything issuing from the software agent as performed by its principal (Allen et al., 1996). In this view, the software agent is downscaled to a simple machine or program such as a phone or an email-program, which cannot automatically conclude a contract on behalf of its principal, but —as shown in figure 2— in this case it is legally assumed that the principal only used the software agent as a medium of exchange to transmit his will and therefore is legally bound by the software agent’s actions. As Lerouge puts it:

[...] if a party creates a situation in which the electronic agent is to act on his behalf, then the party is bound by the actions of the ‘agents’ (Lerouge, 1999).

3. Legal persons are sometimes referred to as juridical persons or artificial persons to distinguish them from natural persons.
This approach has got the advantage that it gives a strong incentive to the principals to control that their software agents operate properly. On the downside, it is very harsh. For example, Allen and Widdison, who favoured the perception of software agents as simple passive communication tools, asked:

Is it fair, or even commercially reasonable, to hold the human trader bound by unexpected communications just because it was theoretically possible that the computer would produce them (Allen et al., 1996)?

This compunction of Allen and Widdison refers to the case that for example the third party is aware that the communication produced by the software agent does not represent the principal’s intentions. If software agents are legally treated as mere communication tools, the third party is in the best position to knowingly take advantage of the software agents “transmission error”. In this case, the principal would have to deal with the consequences, that could result from programming faults, errors of calculation or simple electronic bugs (Weitzenböck, 2001).

Nonetheless, using this legal interpretation, the “entity”-characteristic does not conflict with the capability criterion of the contract conclusion through agents, although the contracting by agents is excluded in that case. Despite the obvious conflict with contract conclusion through and by agents depending on the interpretation of the term entity with regard to legal personhood, the approach of considering software agents as simple communication tools is the most well accepted one by legal authors and was considered by the US and the Canadian legislation (Andrade et al., 2004b) (see section 4 for more details).
3.2. **Social Ability**

The second characteristic of software agents to be analysed is the social ability. Social ability means that software agents are capable of interacting with other (software) agents (and possibly humans) in order to satisfy their design objectives (Wooldridge, 2001, p. 23).

This capability is essential in legal terms as it is a prerequisite for any legal contract. Thus, as already pointed out in section 3.1, both civil and common law define two essential elements for the validity of a contract: capacity and consent (Øren, 2001, Weitzenböck, 2001). Whereas the “entity” characteristic is important in terms of capacity, social ability is mandatory for the consent. Thus, in order to establish a contract, in legal terms the most important feature is that one party makes an offer for an arrangement that another accepts, i.e. each party has to declare its intention to contract (for a given set of contractual terms, defined in the offer of the offeror). This is called a “concurrence of wills” or “ad idem” (meeting of the minds) of two or more parties. Having social ability, software agents have the means to express their declarations of intentions and consequently the basis to enter into contract negotiations and –from the consent point– to conclude contracts in the first place.

3.3. **Reactivity**

Reactivity refers to software agents being able to perceive their environment and respond in a timely fashion to changes that occur in that environment in order to satisfy their design objectives. This important feature differentiates software agents from procedural programs that cannot change while the procedure is executing. Thus, if the environment changes and the underlying preconditions for a contract change as a result of this, the software agent is able to adapt to that situation. Although theoretically this is not required by the law, it is especially relevant with regard to error tolerance. Hence, if the software agent would not be able to adapt to changes it might conclude wrong contracts and as a result it or its principal might be reliable for that error. This however would significantly increase the economic risk associated with software agents interacting in electronic markets and thus would restrain this kind of commerce at large.

3.4. **Pro-Activity**

Pro-activity implies that software agents are capable to exhibit goal-directed behaviour by taking the initiative in order to satisfy their design objectives, i.e. they are driven by a set of tendencies such as a contract goal given by the principal. This intention and power to arrive at a goal or intention declaration are needed for the consent element for the contract validity.
3.5. Autonomy

The last characteristic of software agents that is analysed legally in this paper is the most debated on: “autonomy”. In legal terms, autonomy is closely linked to the “personhood” discussion (i.e. the entity) and therefore should be seen in connection with section 3.1 (Karnow, 1996, Kerr, 1999). Although not meaning explicitly the same, complete autonomy by an entity (i.e. the acting and concluding of contracts by software agents without any interplay with the principal as shown in figure 3) is legally often interpreted as legal personhood.

Figure 3. Autonomous Software Agents with Legal Personhood

As shown in figure 3, with regard to the agent this implies that it is independent of its principal (i.e. no principal can be found in the figure) and acts on its own accord. What is interesting with regard to the figure is that is the same figure as figure 1. This is a result of most existing definitions being agent-centred and lacking the inclusion of a principal an agent might act for.

When looking at the consequences of the attributing this extreme form of complete autonomy to software agents, this would imply to confer legal personhood to them and to then develop a theory of liability on that basis (Andrade et al., 2007). This approach has got at least two advantages. First of all, by the recognition of an autonomous consent the question of the validity of declarations and contracts enacted or concluded by software agents would be solved without affecting the legal theories about consent and declaration, contractual freedom, and the conclusion of contracts too much (Feliu, 2001). Secondly, it would take away from principals a rather big proportion of the risk, because, by considering the eventual liability of the software agents, it would reduce their responsibility for the software agents’ behaviour (Sartor et al., 2002).

This approach seems to be rather convenient in all aspects, nevertheless, its adoption poses difficulties as it is very hard to fit software agents into the corset of legal personality. Miglio et al. annotate for example:
The arguments made for justifying the attribution of legal personality to an entity are based on at least three aspects: moral authority, social capacity and legal convenience (Miglio et al., 2002).

Looking at the criteria, it becomes obvious that it is debatable whether software agents can be classified as juristic persons. Thus, even though Wooldridge’s and Jennings’s notion of software agents assumes intelligence and autonomy (see section 2) it is not at all certain that software agents can achieve self-consciousness i.e. that they make conscious, moral decisions on their own.

The argument of social reality needed for conferring legal personality is based on the fact that many artificial legal persons are regarded as social organisms with a distinct social will. It can be debated in how far the social will can be attributed to software agents, however in literature quite often the view is advanced that the gap to software agents having the capacity for social action is still a substantial one (Allen et al., 1996).

The third argument for justifying the attribution of legal personality to software agents mentioned by Miglio et al. – the legal convenience – has been originally applied to confer personality to limited liability companies. This conferment has some advantages:

- having legal personality a company can sue or be sued in its own name and independently of its members, it can own its own property (independently of its members), and
- can have perpetual succession until dissolution, thereby surviving the death or change of its directors and shareholders (Weitzenböck, 2001).

Even assuming that it is possible to fit software agents into the above mentioned corset, some problems remain. Thus, an initial and direct consequence of conferring software agents with legal personality would be that they would legally be subject to the classical theory of law patrimony (Chopra et al., 2004). Patrimony is composed of both active legal situations (such as rights, powers, options) and passive legal situations (such as obligations, burdens and duties). Hence, in case of ascribing legal personality to software agents this would imply to give them patrimonial rights and to also make them subject to liabilities for negligent acts or omissions, just as humans.

Another argument that is put forward when justifying that it is still difficult to attribute legal personality to software agents, is the problem of the software agent identification: does it coincide with the hardware or with the software? And what happens in the case that software and hardware are dispersed over more that one site owned by different individuals (Allen et al., 1996)? As in the case of companies, a possible solution may be to set up a kind of registry where anyone who wishes to use software agents in e-commerce should register not only the name for identifying the software agent and a kind of digital signature of the agent, but also himself for the purpose of being identified as the software agent’s principal (Allen et al., 1996, Karnow, 1996, Miglio et al., 2002). However, in contrast to the case of limited liability companies mentioned above, in economic terms the expenses for introducing such a system...
might be hard to justify as traders could easily include an interchange agreement in their contracts which would serve the same purposes as a register. That’s why looking at all the arguments, this paper draws the conclusion that at the moment it would be in conflict with existing law to construe autonomy as complete autonomy and on that basis give software agents the status of juristic persons.

If the interpretation of the term “autonomy” is eased to “conditional autonomy” another legal interpretation of the conclusion of contracts by and through software agents can be thought of. This interpretation is shown in figure 4 and is known as the “Law of Agency”.

Figure 4. Software Agents and the Law of Agency

Agency is the fiduciary relationship that arises when one person (the principal) authorises another person (the agent) to create a legal relationship with a third party
on the principal’s behalf. Generally speaking, this means that when agreements are made by the agent and the agent has adverted that he or she is acting on the principal’s behalf, the principal is liable for any contract(s) made by the agent, as long as the agent has done what he or she was instructed to do. Thus, the result is, with some exceptions, the same as if the principal had done it by himself.

At a first glance, applying the law of agency to software agents seems very tempting and has been discussed at length by a great deal of literature in the United States as well as in Europe. After all, one might want to argue that a software agent autonomously concluding contracts for its principal without any human intervention, is very similar to a human agent serving the same purpose. This view has been put forward by Fischer, who argues that the comparison seems obvious:

[...]

Fischer argues his approach to be reasonable, as under the American law of agency, an agent does not require contractual capacity in order to be able to act as an agent. However under the common law, an agent needs to accept his mandate as according to §26 of the Restatement 2d of U.S. Agency Law, in order to establish an agency relationship, the parties have to express a written or verbal agreement that determines the authority, duties and liabilities of the agent (Miglio et al., 2002). Fischer himself admits that the application of this concept to computers is not feasible, and as a consequence proposes the adoption of the legal fiction consent to try to satisfy this criterion of agency law.

Looking at law of agency, it furthermore has to be noted that it can be based on two different kinds of representation, either on representation demanded by the law (i.e. legal representation), or on a voluntary agreement between two parties (i.e. voluntary representation) (Barfield, 2005). In voluntary representation, at least in civil law countries, it is not absolutely necessary that the representative has full legal capacity, but it surely is required that he portrays the “natural capacity of understanding and wanting” (Andrade et al., 2004a) (i.e. that he needs to have a minimum of capacity to understand the configuration and possible consequences of the act to be performed), because the representative acts in the name of the principal and concludes legally binding contracts for him. As Bellia puts it:

A person with limited capacity, such as a minor, may be an agent, but a person with no capacity whatever may not (Bellia Jr., 2001).

Of course one might wonder whether an intelligent software agent might have that minimum of capacity to perform representative acts. No doubt that this kind of devices will probably have a much greater capacity to foresee all the consequences of its acting than any minor. Nevertheless, there is still a difficulty that the application of
the law of agency for software agents cannot solve. One of the biggest problems that is closely linked to section 3.1 is that the law of agency is only applicable for legal persons and only legal persons can make contracts. This flaw was recognised by Kerr who suggested:

[...] in order to invoke the principles of agency, it is therefore necessary to include electronic devices within the set of rules that form the external aspect of agency (Kerr, 1999, p. 55).

Kerr justifies this exclusion of the internal agency relationship (meaning the one between the principal and the software agent) by quoting Bowsteadt and Reynolds (Kerr, 1999), who claimed that disputes in e-commerce will only take place between the principal and the third party. This assumption however leads to the problem of the principal being without any recourse against the software agent in situations where the agent exceeds its sphere of influence or when it employs another incompetent agent. Consequently, the principal is in the situation of having rights and duties with respect to the third party, but not with respect to internal relations (Jurewicz, 2005, Lerouge, 1999).

Besides all the above mentioned problems with the application of the law of agency to software agents, the agency theory has been criticised for its relatively complex and sometimes even incomprehensible structure. De Miglio for example claims that it fails to explain several issues and makes to many exceptions to the law of agency (Miglio et al., 2002).

Summing up, this paper concludes that the application of the law of agency to software agents, though attractive, leaves a number of unanswered questions such as the internal principal-agent relationship. That’s why, although favored by many scientists, in the eyes of the authors it cannot be used for software agents under the current legislation.

4. Legislation on the Contract Conclusion by Software Agents

After having had a look on how the software agents characteristics go along with the current legislation on contract conclusion, in this section, the focus will be on a number of enacted legislations and the way these have dealt with the negotiation and conclusion of contracts by automated means such as software agents. Thereby, the pros and cons of the individual laws and their articles will not be discussed, but the laws as well as their implications for software agents will only be presented as facts.

Some of these laws, such as the UNICITRAL Model Law on Electronic Commerce (UN, 1996) or the U.S. Uniform Electronic Transaction Act (National Conference of Commissioners on Uniform State Laws, 1999b) have taken the approach to legally attribute the operations of the software agents to the principal. The United States Uniform Computer Information Transaction Act (UCITA) (National Conference of Commissioners on Uniform State Laws, 2002) attributes the actions of the software agents to the principal, however, an overtone of the law of agency can be found in
some sections. In contrast, the European Union’s Electronic Commerce Directive (EU, 2000) takes a slightly different approach. It is the only legislation that does not directly refer to software agents, however it directly encourages EU member states to enact legislation that allows for the conclusion of contracts by electronic means. Last but not least the current case law on software agents will be discussed to finish the examination of the legislation on contract conclusion of software agents.

4.1. The United Nation’s UNCITRAL Model Law on Software Commerce

The first legislation that will be briefly looked at is the UNICITRAL Model Law that was passed by the United Nations General Assembly Resolution 51/162 on December 16th 1996 in order to further the progressive harmonization and the unification of electronic commerce. The UNICITRAL Model Law is in so far interesting, as its Article 2 already implied the use of software agents. Thus, article 2(c) (UN, 1996) states that:

“Originator” of a data message means a person by whom, or on whose behalf, the data message purports to have been sent […]

In the Article-by-Article Remarks for article 2 (UN, 1996, article-by-article remarks) further information on the role of the computer program:

Data messages that are generated automatically by computers without direct human intervention should be regarded as “originating” from the legal entity on behalf of which the computer is operated. […]

Hence, in the view of the UNCITRAL, software agents are mere communication tools. The software agent is limited to electronically transmit the declarations of intentions of its principal. As a result, the principal is obligated for all unforeseen events. The underlying idea is that the person using a software agent is liable for the effects deriving from its use, as long as the software agent is not capable of autonomously expressing an own will.

4.2. The European Electronic Commerce Directive

The Electronic Commerce directive is based on the UNICITRAL Model Law on Software Commerce. It is fairly procedural and does not establish any substantive rules of European law. Hence, a European Community (EC) directive is binding, as to the result to be achieved, upon each Member State to which it is addressed, but it is left to the national authorities to choose the form and methods (EC, 2002, §249 III). Nevertheless, Article 9 (1) states that:

the Member States shall ensure that their legal system allows contracts to be concluded by electronic means [and] that the legal requirements applicable to the contractual process neither create obstacles for the use
of electronic contracts nor result in such contracts being deprived of legal effectiveness and validity on account of their having been made by electronic means (UN, 1996).

As in the UNCITRAL Model Law on Software Commerce reference on software agents can be found in the European Electronic Conference. Thus, within the explanatory notes of the Electronic Commerce Directive software agents are mentioned, as it is stated that:

the Member States will have to: [...] not prevent the use of certain electronic systems as intelligent electronic agents (UN, 1996).

In spite of that, according to Jurewicz the “electronic agent” regulation was not included in either the recitals or in the articles of the directive (Jurewicz, 2005).

Scientists have been debating whether the earlier-cited Article 9(1) as well as Article 11 of the European Electronic Commerce Directive which govern electronic contract formation, can be applied for the conclusion of contracts by software agents. A strict interpretation of Article 11 may suggest that the article refers to a human rather than an electronic agent, as it does not allow an automatic electronic response since the language used in the provisions refers to “a recipient” and “his consent”.

On the other hand, it could also be argued that no exceptions which preclude “a recipient” from being an electronic agent are made in the Directive. This approach has been accepted by the majority, because it promotes the development of electronic commerce and complies with international standards (Mirzaian, 2002).

4.3. The United States’ Uniform Electronic Transaction Act

The United States’ Uniform Electronic Transaction Act (UETA) was approved and recommended on July 23rd-30th, 1999, by the National Conference of Commissioners on Uniform State Laws (NCCUSL) to be enacted in all states of the U.S. The law was promulgated to remove electronic commerce barriers by validating and effectuating electronic records, signatures and contracts. Hence, in the “Prefatory Note” to the Act, the drafters stated:

the Act makes clear that the actions of machines (electronic agents) programmed and used by people will bind the user of the machine, regardless of whether human review of a particular transaction has occurred (National Conference of Commissioners on Uniform State Laws, 1999b, prefatory note).

The term “electronic agent” (i.e. software agent in this paper) is then defined in Section 2(6):

“Electronic agent” means a computer program or an electronic or other automated means used independently to initiate an action or respond to electronic records or performances in whole or in part, without review
or action by an individual (National Conference of Commissioners on Uniform State Laws, 1999b).

Accordingly, the UETA directly recognises the autonomy of software agents. This view is intensified by the drafter’s comment number 5 to section 2 of the UETA:

While this Act proceeds on the paradigm that an electronic agent is capable of performing only within the technical structures of its preset programming, it is conceivable that, within the useful life of this Act, electronic agents may be created with the ability to act autonomously, and not just automatically. That is, through developments in artificial intelligence, a computer may be able to “learn through experience, modify the instructions in their own program, and even devise new instructions” (Allen and Widdison (Allen et al., 1996, p. 25)). If such developments occur, courts may construe the definition of electronic agent accordingly, in order to recognize such new capabilities (National Conference of Commissioners on Uniform State Laws, 1999b, comments).

Hence, the drafter of the UETA recognised that software agents may act autonomously, however they considered them as mere tools of their principals and their actions as extension of the human action. This picture is intensified when looking at section 10 of the UETA that provides rules dealing with the effects of errors. The section contemplates several effects of human errors in automated transactions, however no software agent related errors such as a malfunction of the agent, etc. are mentioned.

4.4. The United States’ Uniform Computer Information Transaction Act

Like the UETA, the United States’ Uniform Computer Information Transaction Act (UCITA) was approved and enacted by the NCCUSL in 1999. It was intended to be a response to the enormous growth of electronic commerce at that time and created a framework for electronic contracts. Originally the drafters wanted to incorporate the act into the Uniform Commercial Code (UCC), a former version of the UETA, as Article 2B, but in the progress of the further elaboration decided to enact it as a separate act.

As all other U.S. legislation, the UCC treated software agents as mere communication tools of their principals. However in the subsequent draft of the UCITA, some modifications were made. The final version of the UCITA defines the term “electronic agent” as follows:

a computer program, or electronic or other automated means, used by a person to initiate an action, or to respond to electronic messages or performances, on the person’s behalf without review or action by an individual at the time of the action or response to the message or performance (National Conference of Commissioners on Uniform State Laws, 2002).

Although the phrase ‘on the person’s behalf’ is used, what might indicate some kind of agency relationship, the Official Comments to the UCITA state that:
The legal relationship between the person and the automated agent is not equivalent to common law agency, but takes into account that the “agent” is not a human. However, parties that use electronic agents are ordinarily bound by the results of their operations (National Conference of Commissioners on Uniform State Laws, 1999a).

In Section 202(a) of the UCITA, software agents are then attributed the ability to conclude legal contracts:

A contract may be formed in any manner sufficient to show agreement, including offer and acceptance or conduct of both parties or operations of electronic agents which recognise the existence of a contract (National Conference of Commissioners on Uniform State Laws, 1999a).

As the UETA, in section 213(a) the UCITA attributes the legal actions of the software agents to their principal, which are legally made responsible for the agent.

An electronic authentication, display, message, record, or performance is attributed to a person if it was the act of the person or its electronic agent, or if the person is bound by it under agency or other law (National Conference of Commissioners on Uniform State Laws, 1999a).

Despite this strong notion of treating software agents as simple communication tools enacting the will of their principals, in section 112(b) it can be noted, that parts of the agency concepts are nevertheless included, as direct reference is made to the need that according to §26 of the Restatement 2d of U.S. Agency Law, in order to establish an agency relationship, the parties have to express a written or verbal agreement that determines the authority, duties and liabilities of the agent:

[...] an electronic agent manifests assent to a record or term if, after having an opportunity to review it, the electronic agent authenticates the record or term or engages in operations that in the circumstances indicate acceptance of the record or term (National Conference of Commissioners on Uniform State Laws, 1999a).

This overtone of the law of agency in some sections makes it hard to follow the central idea of the UCITA and in the eyes of the paper made it open to a number of discussions.

4.5. Case Law on Software Agents

After having had a look on the texts of several laws, this last subsection shall deal with the case law on software agents. So far, the case law on software agents is relatively scarce as firstly the application of software agents for e-commerce transactions is still in the early stages of development, secondly courts have only heard a few cases focusing on the conclusion of contracts by software agents, and thirdly many cases that would have solved some of the issues mentioned in this paper are settled (Jurowicz, 2005). In order to show how case law decisions can treat similar situations
differently on an international level, this paper will focus on two cases, an American and a German one. Both cases “only” dealt with automated systems of communication and hence might not be completely appropriate for the analysis of software agents, however as shown in section 3.1, the classification of software agents as mere passive communication tools is the most well accepted one by legal authors and as such has a high significance for the future legal decision on software agents.

The first American case that involved automated systems of communication was Corinthian Pharmaceutical Systems, Inc. vs. Lederle Laboratories\(^4\). In this case the court decided that a response of the seller’s computer and the issuance of a tracking number for a purchase did not present an acceptance of the buyer’s offer. As an explanation the court reasoned that the telephone computer ordering system that was used for the answer, performed automated acts that could not constitute a legal acceptance (Jurewicz, 2005). Observers of the case later argued that the courts decision might have been different in case a more sophisticated system (i.e. one that verified the validity of the incoming orders, checked the inventory level, allocated the necessary part of the inventory for the order and then issued a tracking number) would have been employed.

In contrast, a German court, interpreting the German law, decided differently on a very similar case. The case involved a seller using an automated communication system for offering and selling goods via the Internet. The automated system offered goods below the price intended by the seller and generated automatic replies for the buyers emails saying that the “offer” would be “carried out” immediately (UN, 2003). The court argued that the person on whose behalf the program had been developed and in whose name the messages were sent (i.e. the principal) was legally responsible. Hence, the court recognized the electronic messages as legally binding expressions of one’s intentions that therefore result in valid contracts (UN, 2003). Further case law decisions are discussed in (Brazier et al., 2004) for example.

Just looking at these two court decisions it has to be noted, that at the moment, no standard legislation exists, neither nationally\(^5\), nor internationally. This problem is aggravated as soon as deals are close on an international level, as then the question comes up, which national law to use, especially in the case this has not been agreed on in the general terms and conditions of the contract. As a consequence it is important to solve the issues concerning the conclusion of contracts by software agents, not only on a national level, but to work on an international solution.

\(^4\) 724 F Supp 605, United States District Court, S.D. Indiana, Indianapolis Division, 1989.

\(^5\) In similar cases to the one just mentioned, German courts have already decided differently. That discrepancy is said to stem from conflicting views regarding the distribution of risks in e-commerce transactions (UN, 2003).
5. Remaining Legal Questions

Last but not least, some remaining legal problems shall be discussed, starting with the most important one: the risk bearing in case contracts concluded by software agents fail. Although, as shown in the section 3, according to the different legislation, contracts may be concluded with the help of software agents, by applying the objective theory of contracting that signifies that contracting parties are bound to a contract if they expressed their intention to do so, this is of little use in case possible transaction partners do not feel comfortable with it and refrain from using software agents.

Within the existing legislation, situations are imaginable, where one of the contracting parties wants to get out of a contract concluded by a software agent. To do so, it then has the chance to either argue that the software agent failed, was defective or acted improperly and consequently a mistake or an unfair unconscionability (in common law countries), or an error or bad faith (in civil law countries) occurred (Weitzenböck, 2004).

One way to deal with this problem is proposed by Weitzenböck. She suggests to try to

[...] identify how risk is to be apportioned where one of the parties is alleging that the electronic agent malfunctioned (Weitzenböck, 2004).

However, as she herself points out, it can prove to be rather difficult to decide whether the risk should be borne by the programmer, the principal and/or the person who trained the agent. In case the software agent malfunctions, the user might have remedies against its programmer, but of course the programmer might have limited his warranty or even excluded it in the general terms and conditions of the contract.

Another problem that needs to be addressed is the one of the residual error. According to the British (Beale, 2006) and the U.S. (Corbin, 1982, §105) legislation an offeree cannot legally bind the offeror, in case he knows or has reason to know that the offer has been changed accidentally due to residual errors and does not represent the original intentions of the offeror any more. Thus, in case a “reasonable man” could suspect an occurred residual error, the offeree is not permitted to “snap-in” (Lerouge, 1999). The situation gets more difficult in case the offeree could not have suspected an error and has reasons to believe that the offer made by the offeror corresponds to his true intentions. For this case judicial doctrines are divided. Corbin for example argues that the delivered offer would be binding for both transaction partners and hence the sender of the offer is deemed to take the risk of a residual error and to pay the “cost of confidence” (Corbin, 1982, Cavanillas, 2001), however his opinion is heavily discussed.

Summing this section up, the conclusion can be drawn that issues of security as well as the proper functioning and the reliability of software agents are very important for their users. A number of suggestions have been put forward to address this problem.
One of them was suggested by Stuurman and Wijnands. They proposed a labelling system very similar to that by Lerouge, only that their system wasn’t a voluntary one. In detail, Stuurman and Wijnands proposed the development of a security classification in combination with the certification of agents by reference to a particular class of security standards. This would allow requirements to be imposed in respect to the security level which the software agent must fulfil if its principal wants it to be authorised or accepted for certain activities (Stuurman et al., 2001). As a consequence of such a system, controlling mechanisms (e.g. for a monitoring in order to determine whether the agent complies with the specified level of security) are very likely to be required. However, as mentioned above when discussing Lerouge’s proposal, the economic reasonability has to be proven.

6. Conclusion

Although still not widely used today, it is not unreasonable to predict that in the future, software agents as initiators and mediators of electronic transactions assisting human interaction through all stages of the transaction process, are very likely to foster economic commerce. However, for this to happen to a large extend, the doubts of the potential users have to be allayed. A starting point to strengthen consumers confidence is the general resolution of today’s legal questions, as potential users will almost for certain refrain from using electronic agents in case of legal uncertainties. This is especially relevant since - as Brazier et al. pointed out - “technical progress warrants the need for legal clarification, and the legal implications may require resolution of technical issues.” (Brazier et al., 2004)

This paper focused on the analysis of one of these questions and discussed the legal problems related to the conclusion of contracts by software agents.

Starting from the very general notion of agency in AI, it had a look at the components of the agent definition and discussed them with regard to the current legislation. It thereby pointed out the special problems related to the criteria “autonomy” as well as “entity” with respect to legal personhood as well as two other contractual concepts (i.e. the consideration of software agents as mere passive communication tools and the consideration of the law of agency for software agents). A summary of findings and the discussion in this paper can be seen in table 1. Furthermore, the different current nation and international legislation on the conclusion of contracts by software agents were presented. In addition, the paper outlined a number of ways to deal with today’s legal uncertainties such as establishing a labelling system for agents or developing a certification system for them.
Table 1. Summary

<table>
<thead>
<tr>
<th>Software Agents with Legal Personhood</th>
<th>Basic Idea</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conferment of legal personhood to software agents and thereby attribution of liability</td>
<td>Validity recognition without affecting existing legal theories. Reduction of risk for the principals.</td>
<td>Reduction of the principals' incentives to control that their software agents operate properly. So far only considered in case law decisions.</td>
<td></td>
</tr>
</tbody>
</table>

| Software Agents as Passive Tools | Anything issuing from the software agent as performed by its principal → software agent is only a medium of exchange for the principal and the principal is legally bound by the agent's actions. | Explicit legal regulation. Gives strong incentives to the principal to control that their software agents operate properly. Notion employed by the United Nation's UNCITRAL Model Law on Software Commerce and the United States' Uniform Computer Information Transaction Act for example wrt.: Third party is in the best position to knowingly take advantage of the software agents" transmission errors. |

| Software Agents under the Law of Agency | Fiduciary relationship - when agreements are made by the agent and the agent adverted that he or she is acting on the principal's behalf, the principal is liable for any contract(s) made by the agent, as long as the agent has done what he or she was instructed to do. Thus, the result is, with some exceptions, the same as if the principal had done it by himself. | Combines advantages of the two extreme positions. Not a well known and discussed concept. Complexity. Definition of "legal capacity" of a software agent. |


Looking at table 1, all the presented proposals have their advantages and their disadvantages. Thus, although at the moment the usage of software agents as mere communication tools seems to be the most widely considered approach in legislation, several case law decisions have shown a movement towards the other approaches as well. And also in research first examples of agents that could be employed for law of agency-relations have been presented. Thus, Sartor for example in 2009 published an article in which he described how to attribute agents with cognitive states and how to use those cognitive states in order to apply the concept of legal personality with agents (cognitive automata) (Sartor, 2009). This development is important as agent technology is progressing at a fairly steady pace and legal aspects shouldn’t be neglected. In the view of the author, one good point to continue the discussion is by having a closer look at the agent definition. Thus, as figures 1 and 3 have shown, at the moment the agent definition lacks the consideration of potential principals of the agents. That’s why it rather indirectly seems to support the idea of agents with legal personhood, i.e. an idea which as described before poses several legal problems and is not considered by any legislation to this point. By closing this gap and considering the owners as well as programmers of agents in the agent definition numerous legal issues could be approached and agent technology could overcome one hurdle on its way to broader adoption.

7. References


Mirzaian A. G., « Electronic Commerce: This is Not Your Father’s Oldsmobile », 26 Rutgers Law Record, 2002.


