International Journal of Artificial Intelligence,
Spring 2010, Volume 4, Number S10
ISSN 0974-0635;

Automated vs. Human Negotiation

Michael Filzmoser

Institute of Management Science
Vienna University of Technology
Theresianumgasse 27, 1040 Vienna, Austria
michael.filzmoser@tuwien.ac.at

ABSTRACT

Automated negotiation, in which software agents assume the negotiation tasks of their human users, is argued to improve the benefits of e-business transactions. Higher benefits result on one hand from reduced transaction costs due to the avoidance of human intervention, on the other hand software agents are supposed to achieve better outcomes. While the former argument is straightforward the latter lacks empirical evidence. The few studies that compare human and software agent performance in automated negotiations only come to inconclusive results. We model and simulate automated negotiation systems and compare the output of the simulation runs to outcomes of negotiation experiments with human subjects. The automated negotiation systems consist of software agents that follow concession strategies proposed in negotiation literature and appropriate protocols that allow these agents to interrupt their strategy to avoid exploitation and unfavorable agreements. The negotiation problems used in the simulations are those derived from the experiments so that outcomes of human and automated negotiation are directly comparable. The outcome dimensions considered in our analysis are the proportion of agreements, dyadic and individual performance, and fairness. Only a set of systems managed to significantly outperform human negotiations in all outcome dimensions. These systems consist of software agents, that systematically propose offers of monotonically decreasing utility and make first concession steps if the opponent reciprocated previous concessions. The protocols of these systems enable to reject unfavorable offers to avoid exploitation or unfavorable agreements without immediately aborting negotiations.

Keywords: simulation, negotiation, automated negotiation.

Journal of Economic Literature (JEL) Classification Number: C63, C70, C88, O30.

1 Introduction

The present hype of agent-based methodologies in computer science and the potentials of software agents that interact autonomously over the Internet established the field of automated negotiation, which is supposed to be of considerable importance for research and practice in the future. Though in e-business information, orders, and payments can be handled electronically, automation for the task of negotiating the final contract is still missing to a large extent. Currently e-business transactions still require human intervention, which increases transaction costs and diminishes the potential benefits of electronic business (Maes, Guttman and Moukas, 1999). Scholars argue that the evolution of agent-mediated electronic business will change the future of traditional business and lead to a radical reorganization of economic structures (Kontolemakis, Kanellis and Martakos, 2004).

Clearly automated negotiation will only be applied instead of traditional negotiation between humans if and where there are benefits of doing so (Blecherman, 1999). Such potential benefits of automated negotiation for e-business can either result from lower transaction costs, improved negotiation outcomes, or a combination of both. Concerning transaction costs, automated negotiation doubtlessly reduces opportunity costs compared to traditional negotiations as it surrogates human involvement. Furthermore the direct costs of automated negotiation itself are negligible. Fast progression consumes minimal computer resources which continuously become cheaper or would be idle anyway (Cranor and Resnick, 2000). Regarding the outcome of automated negotiation, many scholars share the assumption that automated negotiation can achieve better outcomes than traditional negotiation as software agents are superior to humans in dealing with complex problems (Oliver, 1996; Sandholm, 1999; Bichler, 2000; Choi, Liu and Chan, 2001). One argumentation in favor of this assumption is that humans – in contrast to software agents – might lack experience and capabilities in negotiating or the willingness to negotiate (Choi et al., 2001; Lomuscio, Wooldridge and Jennings, 2003). Furthermore simulation studies revealed that software agents achieve (nearly) Pareto-optimal results while humans often do not achieve such results in experiments. The higher net benefit due to lower transaction costs and superior outcomes of automated negotiation also could cause a replacement of currently used transaction mechanisms like web catalogs or online auctions (Bichler, 2000; Bichler and Segev, 2001; Lomuscio et al., 2003; Kontolemakis et al., 2004). Besides other reasons, these transaction mechanisms are preferred over traditional negotiation especially due to the higher transaction costs caused by human involvement. Moreover automation enables new types of transactions and the handling of business volumes deemed impossible without as more transaction partners can be reached and dealt at lower cost (Lomuscio et al., 2003; Kontolemakis et al., 2004).

The argumentation concerning reduced transaction costs and higher transaction volumes through automation is straightforward. However, without an actual comparison of the performance of automated negotiation and human negotiation the assumption that automated negotiation outperforms traditional negotiation remains unjustified. Given the youth of the field it is not surprising that operative systems for automated negotiation are not available yet – except a few experimental systems developed for academic purposes (Guttman, Moukas and Maes, 1998). Researchers therefore rely on analytical models, for very specific problems, and simulations, for more complex and realistic settings, to evaluate possible configurations for automated negotiation systems. The assumed preferences, used as an input in simulations of automated negotiations, often are simplistic and do not reflect the possible complexity and variety of the preferences humans actually could have over one and the same negotiation object. The validity of the input to the simulations must be assured especially when evaluating proposed systems configurations for applications in practice and investigating the performance of software compared to human agents. Actually only one study was found in a literature review in which preferences from humans subjects are elicited and used as input for software agents in a simulation of automated negotiation (Bosse and Jonker, 2005). Another study compares for a negotiation case with predefined preferences the negotiation outcomes achieved by software agents in automated negotiations to those achieved by humans in experiments (Oliver, 1996). The results of both above mentioned studies indicate that human negotiators reach better result than software agents or at least equivalent ones.

This paper aims at addressing the lack of empirical studies that actually compare the performance of human and automated negotiation. We implemented automated negotiation systems composed of simple concession strategies, proposed in negotiation literature, and appropriate interaction protocols. Preferences of human subjects from more than 2000 negotiation experiments are used as input to these systems. The outcome of the experiments acted as benchmark for the evaluation of different system configuration. The use of preferences elicited from human subjects in negotiation experiments not only allows to evaluate negotiation systems in realistically complex and manifold settings. For the particular case of the implemented systems it also enables the direct comparison of the results of the experiments and the output of the simulation, as both experience the same input, and therefore the verification of the assumption that automated negotiation can outperform traditional negotiation. The remainder of this paper is structured as follows: Section 2 provides a brief description of the simulation model.[1] Section 3 discusses the experimental design and evaluation criteria. Section 4 presents and discusses the results of the simulation study and compares the performance of automated and human negotiations. Finally Section 5 ends the paper with a summary of the results of the study and the conclusions we draw.

2 Simulation Model

In automated negotiations autonomous software agents conduct negotiations, in performing negotiation tasks and making necessary decisions, based on their users' preferences in following their strategies. The interactions between the software agents are governed by a protocol. The combinations of software agents representing the parties in the negotiation and an interaction protocol specifies an automated negotiation system that automates the negotiation process (Rosenschein and Zlotkin, 1994). Furthermore the negotiation problem – in form of the users' preferences over the negotiation object – has to be indicated as an input to the automated negotiation system, as depicted in Figure 1 (Jennings, Faratin, Lomuscio, Parsons, Wooldridge and Sierra, 2001).

[1] Detailed descriptions and the source code of the simulation program are provided by the author upon request.

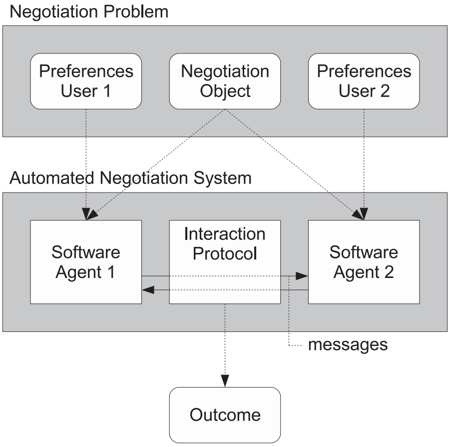

Figure 1: Components of automated negotiation

2.1 Input

We used the preferences elicited from subjects in 2,065 negotiation experiments as input for the automated negotiation systems of our study and the outcomes of these experiments as benchmark for their evaluation. The experiments were conducted by the InterNeg research group with the web-based negotiation support system *Inspire* (Kersten and Noronha, 1999).[2] The case used in the experiments was a bilateral buyer-supplier negotiation about the purchase of bicycle parts. The subjects representing the buyer and the seller negotiate over four issues (price, delivery, payment, and warranty) of a supply contract. In each issue discrete settlement options are available that combine to a total of 180 alternative packages the parties can propose as offers and agree on.

Inspire elicits the preferences over the negotiation object using a hybrid conjoint approach (Kersten and Noronha, 1999): First subjects distribute 100 points over the issues to indicate their importance, then points have to be assigned to the options within each issue to indicate partial utilities. This information is used by the system to calculate the overall utility of several packages for holistic evaluation by the subjects. In case subjects make changes to this overall utilities of packages the preference information is adjusted by means of an ordinary least squares regression. Preferences are used to evaluate the focal negotiator's and its opponent's offers. Utilities values are attached to all offers as well as graphically represented in a history graph that shows the development of the utility of offers as a function of time.

2.2 Software Agent Strategies

Software agents mainly embody a strategy that determines the opening offer and the procedure for generating subsequent offers, the reactions on messages of the opponent, as well

[2]We thank the InterNeg research group for providing access to *Inspire* database.

as acceptance and termination criteria. For simulations we implement rule-based and rather deterministic concession strategies proposed in negotiation literature. These strategies neither model their opponent nor try to learn something about their opponent's preferences or strategy, but are therefore (re)usable for various negotiation problems with different opponents. Existent learning strategies for automated negotiation are severely limited in the scope of parameters they can learn, like for instance stylized strategies or the reservation level of the opponent, and therefore not applicable to the settings of the current study.

Called the first time a software agent proposes its most preferred offer as opening offer and thereafter follows different offer generation strategies that implement the concession-based negotiation approach. The strictly monotonic offer strategy *(SMO)* was proposed by Contini and Zionts (1968). It continuously proposes offers of least possible lower utility and never two offers of the same utility level. The monotonic offer strategy *MON* was observed by Kelley (1966) in negotiation experiments. He argued that this 'systematic concession making model' is successful in reaching favorable outcomes in negotiations. The demand utility level is least possibly lowered only if all offers of the current utility level were already proposed. The least-cost-issue offer strategy *(LCI)* was proposed by Mumpower (1991) as a simple heuristic negotiators could follow in offer generation. Here concessions are successively made on that issue for which a concession costs least in utility. All offers deviating from the last offer in only one issue are considered and the one with the least possible lower utility is proposed next. The lexicographic offer strategy *(LEX)* bases on the concept of lexicographical ordering of alternatives, which was proposed as an alternative to multi-attribute utility ordering in decision making by Beroggi (2003). All possible offers are ordered lexicographically such that offers with options of higher partial utility in issues of higher importance are ranked first. Then the first offer from this list that constitutes a concession is chosen as next offer to propose.

These offer generation strategies are combined with one of two concession strategies that determine, based on the course of the negotiation, whether or not further concessions should be made. This intermediate step provides software agents with the possibility to interrupt offering to avoid exploitation or unfavorable agreements. We follow the approach for fair concession making proposed by Bartos (1977) in his 'simple model of negotiation'. To avoid exploitation negotiators should reject to make further offers if the opponent's concessions are unfairly small and wait until the opponent catches up.

In our model a concession is considered unfairly small if the perceived concession magnitude of the opponent is lower than the focal negotiators concession magnitude. Where the concession magnitude is the difference between the opening offer and the current offer. Two benchmarks for this comparison of concession magnitudes are available and both are implemented in our simulation. The opponent's concession magnitude could be compared to the concession magnitude determined by the last offer of the focal negotiator *(act)*. This is a rather optimistic and active approach as new offers are made as soon as the opponent reciprocated previous concessions i.e. concession steps ahead are made following this strategy. The second alternative is to compare the opponent's concession magnitude to the concession magnitude of the next offer to be proposed *(pas)*, which is a rather pessimistic and passive approach. The generated next offer is only proposed if thereby the focal negotiator stays at or below the

concession magnitude of the opponent.
If software agents perceive the state of the negotiation as no basis for further negotiating according to their concession strategy they interrupt offering – and thereby concession-making – if this is enabled by the protocol. If the protocol allows to send *reject* messages (*protocol 3*) the agent will do so and elicit a new offer from the opponent which could allow to continue negotiations. However, if it is not possible to reject offers and thereby interrupt offering temporarily the software agent will try to interrupt it permanently to avoid exploitation and unfavorable outcomes by sending a *quit* message, which breaks off the negotiations (*protocol 2*). If the protocol neither allows to send *reject* nor *quit* messages (*protocol 1*) the opponent cannot interrupt offering and has to send offers until an agreement is reached. Otherwise, in case the software agent comes to the conclusion that the current state of the negotiation builds the basis for further negotiating, it has to decide whether to accept the opponent's last offer or make the counteroffer just determined. The criterion for accepting an offer of the opponent is simple and equal for all agents: Software agents accept the last offer of the opponent if it affords same or higher utility compared to the next offer to be proposed by the focal agent.
The four offer generation strategies combined with the two concession strategies result in eight strategies for the software agents. The ninth software agent implements a Tit-for-Tat strategy *(TFT)* proposed by Shakun (2005). Here offer generation and concession strategy cannot be distinguished as they are aggregated to an overall strategy. TFT fully reciprocates perceived concessions received by the opponent's last offer, it further negotiates only if the opponent makes a concession and also reciprocates reject messages.

2.3 Interaction Protocol

Obviously concession strategies could be exploited easily if they are not provided with means to interrupt offering. We therefore focused on protocols that enable the temporary or permanent interruption of the strategy the software agent otherwise would follow. After the software agents have been provided with the negotiation object (as common information) and the individual preferences (as private information) the negotiation protocol randomly chooses one agent to send its opening offer. Subsequently software agents alternate in taking their turns in which they send messages. In their turn software agents can send one out of a set of messages determined by the protocol: either *offer*, *reject*, *agree*, or *quit*.
An *offer* constitutes a proposal for settling the negotiation. As such it has to provide options for all issues of the negotiation object, i.e. it has to be a full package offer. Negotiation literature emphasis the opportunities to reach mutually beneficial agreements through package offers, which allow to trade issues of lower importance for issues of higher importance in a logrolling procedure. When a software agent sends a *reject* message it does not propose a new offer, or make any other changes to the current state of the negotiation, but insist on its last offer. This message was proposed as a move in non-cooperative bargaining games (Harsanyi, 1956) and as an approach to avoid exploitation and unfairly small concessions of the opponent (Bartos, 1977). Messages of this type enable the software agents to discontinue offering without terminating the negotiation. A transmission of an *agree* message indicates that the software agent accepts the last offer of the opponent as settlement of the negotiation. Like

in negotiations between humans, which do not need to end with an agreement, an interaction protocol can allow the software agents to send a *quit* message (Beam and Segev, 1997). The interaction protocol stops negotiations without agreement if a *quit* messages is sent by either agent, which therefore can be used to break off negotiations permanently.

The interaction protocol terminates the negotiation either if (i) a software agent sends an *agree* message, or (ii) a *quit* message, or finally (iii) if two subsequent messages of the two software agents were *reject* messages. This last termination criterion is applied to avoid infinite negotiations without progress towards any outcome. If a software agent sends a *reject* message it does not change its internal state, which means that the same message will also be sent in its next turn unless a message of the opponent causes state changes. Consequently if both agents send *reject* messages subsequently there will be no state changes and progress in the negotiation any more and the protocol terminates negotiations.

The combination of enabled messages leads to the three protocols used in our simulation. As can be derived from the above discussion of the messages, *offer* and *agree* messages are mandatory components of any interaction protocol. However, the others are optional and used in different combinations resulting in the three protocols considered in our study. *Protocol 1* allows only to propose and accept offers, *protocol 2* additionally enables the software agents to send *quit* messages and *protocol 3* allows to reject offers besides sending and accepting them. A fourth protocol enabling both *reject* and *quit* messages is omitted as the results would be the same as under *protocol 3*. The results of these protocols are equivalent as software agents always try to temporary interrupt offering to keep on negotiating and eventually reach an agreement instead of permanently breaking off the negotiations if both messages, *reject* and *quit*, are permitted.

Figure 2.3 provides a flow chart of the automated negotiation procedure that schematically illustrates the offer generation and concession making decisions of a software agents and the execution of actions dependent on the applied interaction protocol. The described protocols and strategies were not used, in this combination and with the preference information of negotiation experiments as input, yet. To gain insights about the influence of system configurations on different outcome dimensions in this complex setting simulation of these automated negotiation systems is necessary.

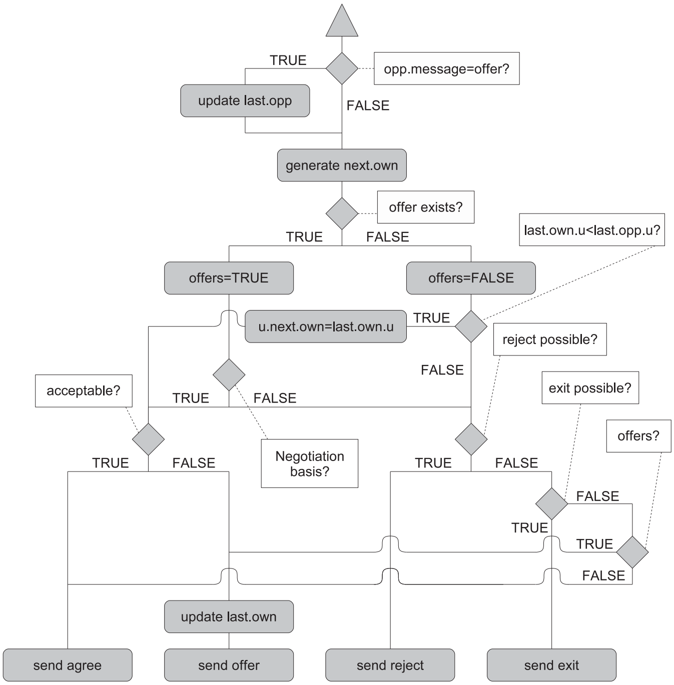

Figure 2: Flow chart of automated negotiation procedure

3 Experimental Design and Measurement

Each automated negotiation system configuration – i.e. each combination of a protocol, an agent representing the seller, and an agent representing the buyer – is one treatment in our experimental design. The three levels for the factor interaction protocol and the nine levels for each of the two parties' software agents results in a total of 243 treatments. A simulation run is fully parametrized if the automated negotiation system and the negotiation problem – i.e. the preferences of the two parties over the negotiation object as input to the software agents – are determined. For each treatment we use the preferences elicited from human subjects in the 2,065 experiments as input to evaluate the system configuration in different settings. This results in 501,795 unique simulation runs. Each simulation run was replicated three times to account for stochastic effects and average results are used for further analyses. A pretest indicated that average results do not significantly differ after three replications any more. Stochastic effects are relevant in the simulation as the starting agent and the next offer to propose in case of ties are determined randomly.

For a holistic evaluation of the negotiation outcomes we apply several measures that complementary evaluate different aspects of the outcome. The consideration of different outcome measures, like prospects of agreement, efficiency, individual utility or fairness, in addition to the aspired holistic evaluation of possible system configurations, enables the selection of system configurations that best support the achievement of specific goals of a given negotiation. Concerning the effectiveness of negotiations we consider the proportion of agreements reached in a treatment. Based on the configuration of the accepted offer the utility of the agreement for the parties can be calculated. As not all agreements are equally good this information is used in further analysis to assess the quality of the agreement. For the evaluation of the dyadic performance of negotiators and automated negotiation systems we use the Pareto-optimality criterion i.e. we determine for the treatments the proportion of Pareto-optimal agreements. An agreement is Pareto-optimal if there exists no other possible solution to the negotiation problem that dominates the agreement reached – i.e. that provides higher utility to one party without making the other party worse off. Pareto-optimality distinguishes only between agreements that exhibit this property or not. However, just like not all agreements are equally good, not all agreements that are not Pareto-optimal are equally bad, but some of the possible solutions can be closer to the Pareto frontier than others. Therefore we calculate for an agreement the minimal Euclidean distance to the Pareto frontier. As mentioned in Section 2.1 the negotiation object consists of 180 discrete solutions. The Pareto frontier therefore is the subset of discrete Pareto-optimal possible agreements. The minimal Euclidean distance to the Pareto frontier is calculated by selecting the minimum of the Euclidean distances in utility space between the reached agreement and the possible solutions that constitute the Pareto frontier. One further measure of joint performance often considered is the fairness of the agreement. We measure fairness in terms of the contract imbalance, defined as the absolute value of the difference of the parties utility of the agreement. The smaller the contract imbalance of an agreement the fairer is it. As these former negotiation outcome measures evaluate the quality of agreements at a dyadic level only, we also consider the utility of an agreement for the two parties as an additional outcome measure for individual performance.

4 Results and Discussion

For the overall data of all treatments the results for the considered outcome measures are mixed concerning the assumption that software agents can outperform humans in negotiations. Table 1 presents descriptive statistics and the proportion of (Pareto-optimal) agreements for the two scenarios human negotiation and automated negotiation. Human negotiation refers to the outcomes of the negotiation experiments while automated negotiation represents the overall results of all treatments in our simulation. One can derive from this table that that our automated negotiation systems seem to reach less (Pareto-optimal) agreements and more imbalanced agreements (lower fairness). On the other hand agreements in automated negotiation are closer to the Pareto frontier and the sellers utility of reached agreements is higher than in human negotiation, while it is fairly the same for the buyer side.

	human negotiation				automated negotiation			
	min. distance to frontier	utility to seller	utility to buyer	contract imbalance	min. distance to frontier	utility to seller	utility to buyer	contract imbalance
minimum	0.00	24.00	26.00	0.00	0.00	20.01	18.45	0.00
1^{st} quartile	0.00	57.00	58.40	7.86	0.00	58.67	56.00	9.69
median	1.00	69.57	70.00	16.67	2.00	71.67	68.74	19.33
3^{rd} quartile	7.07	80.77	80.00	29.17	6.36	84.44	81.05	32.67
maximum	17.49	100.00	100.00	60.73	15.91	100.00	100.00	67.13
mean	5.24	67.93	67.42	20.40	4.29	70.31	67.48	23.65
standard dev.	8.80	18.22	17.28	16.58	6.02	18.18	18.88	18.87
agreements	69.78%				63.41%			
(efficient)	(34.24%)				(32.80%)			

Table 1: Descriptive statistics for the dependent variables in both settings

As these overall effects could caused by some treatments performing worse than others we conduct multiple pairwise comparisons between the 243 treatments and human negotiation as control for all six outcome measures. To be more precise, we test for all outcome measures the one-sided hypothesis that our automated negotiation systems achieves better outcomes than humans in negotiation experiments. Better means different things for the different outcome measures. The alternative hypothesis is that the outcome is greater in automated negotiation than in human negotiation for proportion of (Pareto-optimal) agreements as well as seller and buyer utility of the agreement. For the minimal Euclidean distance to the Pareto frontier and contract imbalance better means that the results achieved by our automated negotiation systems are lower than those achieved in the experiments with human subjects.

As the samples are not normal distributed we use non-parametric tests. For testing differences in minimal Euclidean distance to the Pareto frontier, utility of the agreement to the seller and buyer, and contract imbalance we use the Wilcoxon rank sum test. Differences in proportions of agreements and Pareto-optimal agreements are tested with Pearson's χ^2 test of independence. In multiple statistical hypothesis tests the problem of alpha-error inflation arises – i.e. an increased probability to incorrectly reject the null hypothesis – as a set of hypothesis is tested simultaneously on the same data. To prevent this we control for the family wise error rate in adjusting the p-values by the simple and conservative Bonferroni-Holm method.

For the same negotiation problems as input to the automated negotiation systems, at the significance level $p < 0.05$, 130 of the 243 treatments (53.50% of all treatments) reached a higher proportion of agreements than humans did in the experiments, 90 (37.04%) treatments reached a higher proportion of Pareto-optimal agreements, and the distance to the Pareto-efficient frontier was smaller in 79 (32.51%) treatments. Furthermore the utility of agreements to the seller was higher than the utility human sellers achieved for 154 (63.37%) system configurations, but higher for the buyer side only in 127 (52.26%) system configurations. Contract imbalance was smaller in automated negotiation – and therefore fairness larger – in 112 (46.09%) of the 243 treatments. We derive from these results that not all proposed automated negotiation systems outperform human negotiations. Detailed analyses of the results also indicated

treatment	protocol	seller agent	buyer agent	agreements (%)	efficient agreements (%)	min. distance to frontier	utility to seller	utility to buyer	contract imbalance
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
183	*3*	*MONact*	*MONact*	79.82***	66.44***	0.86***	73.71***	70.86*	8.12***
184	*3*	*MONact*	*MONpas*	74.04*	61.50***	0.86***	74.17***	71.61***	8.07***
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
192	*3*	*MONpas*	*MONact*	75.24***	62.79***	0.89***	74.22***	71.14***	8.06***
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
control				69.78	34.24	5.24	67.93	67.42	20.40

*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$

Table 2: Partial results of multiple-pairwise comparisons treatment vs. control

that not the same systems outperform human negotiation in all outcome measures but that trade-offs exist in system design. Systems achieving good results and superior ones to those reached by humans in one outcome measure often achieved bad results and inferior ones compared to humans in other dimensions. One specific trade-off we also find for the design of automated negotiation systems is well-documented in negotiation literature and commonly found in empirical studies, namely the negotiation dilemma: Factors increasing the prospects of reaching an agreement (proportion of agreements) undermine the quality of agreements reached (minimal Euclidean distance to the Pareto frontier, contract imbalance, individual utilities of the agreement). Furthermore a system achieving agreements of higher utility for one party normally reaches lower utility for the other party and therefore higher imbalance, which adds a further design trade-off so that it becomes even more difficult for systems to succeed in all outcome measures.

However, despite these trade-offs three systems presented in Table 2 significantly outperformed human subjects' performance in the negotiation experiments in all six outcome measures ($p < 0.05$ at least). These systems achieve between five and ten percent more agreements than human negotiation and between 25% and 30% more Pareto-optimal agreements. Furthermore agreements reached are on average about five utility points better for both the seller and the buyer party, closer to the Pareto frontier by five points and more balanced by twelve points than agreements reached in negotiation experiments on the identical set of negotiation problems. Note that the configuration of these three automated negotiation systems that outperform human negotiation reveals an interesting pattern. To outperform human negotiation in all six outcome dimensions and for the negotiation problems used, a system in our simulation model has to operate under *protocol 3* and consist of *MON* software agents only, that follow the monotonic offer strategy in generating offers. Moreover at least one software agent has to follow the active concession strategy *(act)* and therefore make first concession steps if the opponent reciprocated previous concessions.

5 Conclusion

In this paper we presented the results of a simulation study of automated negotiation with the aim to address the assumption that automated negotiation can outperform human negotiation. For this reason we implemented different configurations of automated negotiation systems. The systems consist of software agents that follow concession strategies proposed in negotiation literature and appropriate interaction protocols. As input for these systems we used the preferences elicited from human subjects in negotiation experiments. The simulation output was compared to the outcomes of the negotiation experiment as benchmark for the evaluation of different system configuration.

The analysis of the results of the simulations indicated important trade-offs in the design of automated negotiation systems for the considered negotiation outcome dimensions. Given these fundamental trade-offs it is not surprising that most system configurations only managed to be superior to the benchmark in some outcome measures while they were inferior in others. Only a specific set of systems significantly outperform human negotiations in all six outcome measures. These systems consist of an interaction protocol that enables to reject unfavorable offers and software agents that systematically propose offers of monotonically decreasing utility. Moreover least one of them has to make first concession steps if the opponent reciprocated previous ones. The negotiation process resulting from such a system configuration features a systematic offering of all possible solutions with decreasing utility to the party making the offer. The protocol enables to reject offers of the opponent that provide too low concessions – without immediately terminating the negotiation – so that an agent can wait until the opponent's concession magnitude catches up. This shifts reached agreements in regions of higher utility for both parties and closer to the Pareto frontier resulting in higher efficiency, at both the individual and dyadic level, and fairness. And in case the negotiation get stuck there is one agent that makes concession steps ahead, if the opponent reciprocated previous ones, which increases the prospects of reaching an agreement and therefore the effectiveness of the negotiation.

Obviously the study faces a number of limitations – like the use of student subjects in the experiments or the application of only one negotiation case in both the experiments and the simulation – and a number of extension feasible easily – like improving the software agent strategies with learning and argumentation mechanisms – establishing a wide range of opportunities for future research. However, besides limitations and possible extensions, benefits of automated negotiation could even be exploited with an operative system adopting the simulation system and system configuration of the present study.

For the success and actual application of automated negotiation systems in practice, however, the acceptance of possible users is critical. We actually only considered benefits in terms of superior outcome and lower transaction costs. Other sources of costs, like the loss of power – due to better information, patience, or creativity –, maybe the lost pleasure of negotiating and competitive interaction with other people, the difficulty of preference elicitation, and efforts spent on the configuration or even programming of a software agent, must not be neglected. Humans even could be more satisfied with worse outcomes, compared to those possible with automated negotiation, they reached themselves, due to the effort spent on negotiating and the feeling to have done the best they could, or if social aspects like trust or relationship between the parties are important. These aspects need to be addressed in future research – going beyond questions of mere performance, but tackling questions of acceptance, satisfaction, and usage – to make the insights from simulation studies applicable for the development and implementation of operative systems. This, not the mere simulation of automated negotiation, has to be the final purpose of our endeavor as only the implementation of operative systems will realize the prospected benefits of automated negotiation.

References

Bartos, O. J. 1977. Simple model of negotiation, *Journal of Conflict Resolution* **21**(4): 565–579.

Beam, C. and Segev, A. 1997. Automated negotiations: A survey of the state of the art, *Wirtschaftsinformatik* **39**(3): 263–267.

Beroggi, G. E. G. 2003. Internet multiattribute group decision support in electronic commerce, *Group Decision and Negotiation* **12**(6): 481–499.

Bichler, M. 2000. *Protocols for multi-dimensional negotiation support*, Vienna University of Economics and Business Administration, Vienna.

Bichler, M. and Segev, A. 2001. Methodologies for the design of negotiation protocols on e-markets, *Computer Networks* **37**(2): 137–152.

Blecherman, B. 1999. Adopting automated negotiation, *Technology in Society* **21**(2): 167–174.

Bosse, T. and Jonker, C. M. 2005. Human vs. computer behavior in multi-issue negotiation, *in* T. Ito, H. Hattori, T. Matsuo and M. Zhang (eds), *Proceedings of the First International Workshop on Rational, Robust, and Secure Negotiation in Multi-Agent-Systems*, pp. 10–25.

Choi, S. P. M., Liu, J. and Chan, S.-P. 2001. A genetic agent-based negotiation system, *Computer Networks* **37**(2): 195–204.

Contini, B. and Zionts, S. 1968. Restricted bargaining for organizations with multiple objectives, *Econometrica* **36**(2): 397–414.

Cranor, L. F. and Resnick, P. 2000. Protocols for automated negotiations with buyer anonymity and seller reputations, *Netnomics* **2**(1): 1–23.

Guttman, R. H., Moukas, A. G. and Maes, P. 1998. Agent-mediated electronic commerce: A survey, *Knowledge Engineering Review* **13**(2): 147–159.

Harsanyi, J. C. 1956. Approaches to the bargainig problem before and after the theory of games: A critical discussion of Zeuthen's, Hick's, and Nash's theories, *Econometrica* **24**(2): 144–157.

Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Wooldridge, M. and Sierra, C. 2001. Automated negotiation: Prospects, methods and challenges, *Group Decision and Negotiation* **10**(2): 199–215.

Kelley, H. H. 1966. A classroom study of the dilemmas in interpersonal negotiations, *in* K. Archibald (ed.), *Strategic interaction and conflict*, University of California Press, Berkeley, pp. 49–73.

Kersten, G. E. and Noronha, S. J. 1999. WWW-based negotiation support: Design, implementation, and use, *Decision Support Systems* **25**(2): 135–154.

Kontolemakis, G., Kanellis, P. and Martakos, D. 2004. Software agents for electronic marketplaces: Current and future research directions, *Journal of Information Technology Theory and Application* **6**(1): 43–60.

Lomuscio, A. R., Wooldridge, M. and Jennings, N. R. 2003. A classification scheme for negotiation in electronic commerce, *Group Decision and Negotiation* **12**(1): 31–56.

Maes, P., Guttman, R. H. and Moukas, A. G. 1999. Agents that buy and sell, *Communications of the ACM* **42**(3): 81–91.

Mumpower, J. L. 1991. The judgement policies of negotiators and the structure of negotiation problems, *Management Science* **37**(10): 1304–1324.

Oliver, J. R. 1996. A machine-learning approach to automated negotiation and prospects for electronic commerce, *Journal of Management Information Systems* **13**(3): 83–112.

Rosenschein, J. S. and Zlotkin, G. 1994. *Rules of encounter*, MIT Press, Cambridge.

Sandholm, T. 1999. Automated negotiation, *Communications of the ACM* **42**(3): 84–85.

Shakun, M. F. 2005. Multi-bilateral multi-issue e-negotiation in e-commerce with a tit-for-tat computer agent, *Group Decision and Negotiation* **14**(5): 383–392.