

Available online at www.sciencedirect.com**SciVerse ScienceDirect**

Procedia Computer Science 18 (2013) 1525 – 1534

Procedia
Computer Science

International Conference on Computational Science, ICCS 2013

Complex negotiations in the conclusion and realisation of the contract

Michał Niedźwiecki^a, Krzysztof Rzecki^b, Krzysztof Cetnarowicz^a^aAGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Krakow, Poland^bCracow University of Technology, ul. Warszawska 24, 31-155 Krakow, Poland

Abstract

This paper presents a complex negotiations model that has been adapted to the organisation of the process of the conclusion and execution of a contract between customer (buyer) and bidder (seller), each represented by software agents. Most research on agent-based automated negotiation is focused on models that end their action at the time at which the contract is concluded. However, the implementation of the contract can be a long process. In this time, the needs and capabilities of both sides are subject to change. In complex negotiations, the negotiation is spread over the entire lifespan of the contract and includes pre-contract negotiations as well as renegotiations during the execution of the contract. With this approach, participants can adapt dynamically to changing conditions, and can renegotiate contract conditions which have previously been concluded by both parties.

Keywords: complex negotiation; renegotiation; MAS; contract

1. Introduction

Negotiations are two-sided conversations designed to resolve conflict and to reach an agreement acceptable to both parties. Negotiations are used in many fields, such as trade, logistics, politics, automation, and information technology. Negotiations are made between individual people or institutions, as well as between electronic devices. In trade, negotiations allow the seller and the buyer to reach an agreement whereby both sides are satisfied with the terms of the purchase of goods or services that they have negotiated. In computer science we often have to deal with automated negotiations, which aim to optimise the business process, in order to select the best transmission parameters of the network link, or to help in human decision-making etc. The terminology used in this article is the seller, who offers goods or services, and the buyer, who is in need of goods or services. These goods or services are exchanged for money.

There are many of models of automatic negotiations which solve the problem of reaching agreement between both parties to the negotiations. These have two main drawbacks. The first is that negotiations end as soon as the contract is concluded, and what takes place during the execution of the contract is not serviced. However, after concluding the contract, the preferences of both parties may dramatically change. There may be various reasons for such changes: changes in the market situation, failures and downtimes, or the client simply changes

*Corresponding author: Michał Niedźwiecki; tel.: +48-694-66-11-67.

E-mail address: nkg@agh.edu.pl.

his mind. More advanced negotiation models attempt to predict such situations, and include potential changes in the preferences of both parties. Such negotiation strategies often make the process of concluding a contract longer, which is their other major drawback. Moreover, there will always be a degree of risk involved in predicting future results.

We wanted to develop a model that does not have these disadvantages. In this paper, we describe a solution called complex negotiations. This solution extends the negotiation process across the whole lifetime of the contract. Before concluding the contract, common negotiations are carried out. These use existing models of automatic negotiations. However, after concluding the contract, both parties are able to undertake renegotiations, if they want to change the conditions of the contract. Using complex negotiations, the party that changes its preferences is able to negotiate changes in the contract terms so that they are beneficial to both parties. Renegotiation could be carried out several times. In this way, simpler and faster negotiation strategies could be used.

One drawback of complex negotiations may, in some situations, be the high costs of changing the terms of the contract. However, in environments in which such situation are common, more complex negotiation and renegotiation strategies could be used. In this way, contract terms are changed only where it is really necessary.

Complex negotiations is presented from an agent-based perspective. However we use only two kind of agents (buyer and seller) that could be exchanged for any object that is capable of communication and decision making, such as servers, routers etc.

2. Related works

The terms of the contract that are concluded between the seller and the buyer are agreed in the course of automated negotiation between agents representing both parties. The main areas of research on agent-based automated negotiation are the negotiation protocol and negotiation strategy [1][2].

The protocol defines how to negotiate in order to achieve agreement between the negotiating agents. The basis for automated negotiation protocol multiple-protocol environments is Contract Net Protocol (CNP) [3], adopted by FIPA (the Foundation for Intelligent Physical Agents), and the IEEE Computer Society standards organisation [4]. CNP protocol defines how messages between the two negotiating agents are exchanged. The Initiator commences negotiations, and the Participator is the part that agrees to negotiate. These negotiations are one-to-one. In the CNP, one of the agents tenders for a contract to another agent. Then the second agent rejects or accepts it. An extension framework is used to transmit a counter CNP, which is the third option in addition to acceptance or rejection of [5] OPAC. In addition to one-to-one negotiation there are models of one-to-many and many-to-many. They are implemented as simple one-to-one negotiations to be carried out concurrently in separate threads with each participant in the negotiations. CNP protocol is a basis for other negotiation protocols in systems B2B [6][7], electronic commerce [8], electronic communication (negotiation of transfer[9], and audio/video stream compression[10]).

A negotiation strategy determines when negotiations begin, how they proceed and when they finish. Different negotiation strategies use various models of decision making. Such a model can operate on the basis of a utility function [11], game theory, feature concessions, fuzzy logic [12], neural networks [13], evolutionary programming [14] and other methods such as the Stackelberg game [15]. Research is carried out on strategies that rely on an analysis of market trends and possible future demand, and then choose the best moment to buy [16]. Other methods involve the collection strategy for clients in coalitions [17][18]. The coalition is able to obtain goods in wholesale quantities at a lower cost per unit, so customers in a coalition are in a better situation than if they had had to raise the goods separately. Yet further strategic methods employ concerns such as emotions [19] and arguments [20] with the goal of achieving an outcome which is more satisfactory to both parties. These methods extend the time of a negotiation strategy. In addition, all the methods previously described end at the time of purchase (the conclusion of the contract), and everything that happens subsequently is not considered. However we can adopt it in complex negotiations.

The contract consists of two main phases: the conclusion phase and the execution phase (Fig. 1a). Negotiations are conducted during the conclusion phase, the result of which is the conclusion of a contract, or a break in negotiations. The execution phase starts when the contract is signed, and ends when the contract is fulfilled (finalised). During this phase, the capabilities and needs of both parties regarding the implementation of the

signed contract may change. New sellers with much more attractive offers may emerge, or new customers may increase demand. As a result of these changes it may be necessary to renegotiate the terms of the contract [21] (Fig. 1b). Renegotiations are used where you need to provide QoS, such as ATM networks [22]. Renegotiations are also used in the data stream, such as renegotiation of audio/video stream compression in relation to changing bandwidth for mobile communications [23]. In [24], negotiations and renegotiations layers for Web Services were proposed. In this article we have tried to generalise renegotiations and apply them to the implementation of a system for carrying out the dynamic negotiation of changes in the execution of a contract.

In complex negotiations, we combined the conception of renegotiations with the model of automatic negotiations and we applied it to the process of conclusion and execution of the contract.

3. The concept of complex negotiations

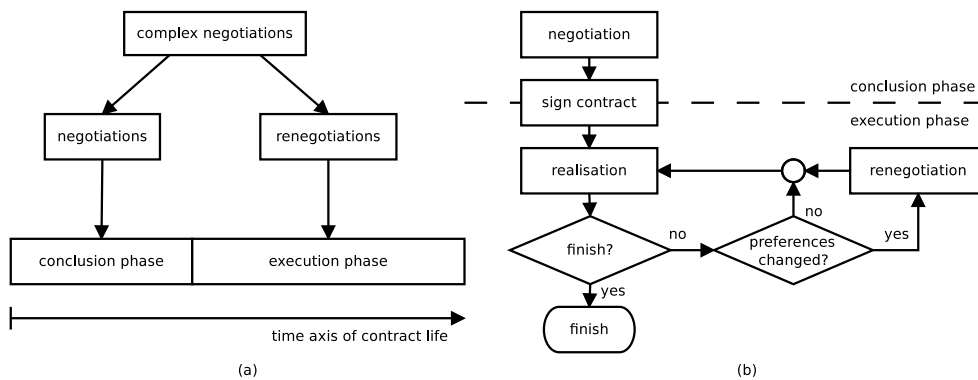


Fig. 1. Contract flow: (a) complex negotiations and contract phases; (b) flow chart of contract execution.

Complex negotiations take place over the lifetime of the contract *C*, and include both the conditions of new contract negotiations, and the renegotiation of conditions which have already been established during the execution phase of the contract (Fig. 1a).

A contract is an agreement between the client (the buyer or the recipient) and the provider (vendor or service provider). For example in [35] emergency dispatcher signs contract with ambulances. The provision of goods or service by the bidder, in return for payment from the client, is the subject of the contract. The roles of clients and providers can be played by servers, software agents [25], etc.

As mentioned above, the lifespan of the contract consists of two phases: the conclusion phase (setting the conditions of contract execution) and the execution phase (contract execution until the fulfilment of the obligations). It is an extension of the arrangement phase and the execution phase of the Entish system [26].

3.1. Negotiations and conclusion phase of the contract

The conditions of the new contract are the subject of negotiations. Negotiations are carried out according to a protocol based on the CNP. The role of initiator (negotiations starter) is always played by the buyer, and the role of participant (the other party in the negotiations) by the seller. If during the negotiations the two sides reach an agreement, then the contract is concluded and they move on to the execution phase of the contract.

Negotiations are carried out according to the following algorithm (shown in Fig. 2a):

1. **initiator**: reports willingness to begin negotiations by sending a message `call for offer`.
2. **participant**: decides one of:
 - (a) break of negotiations, by sending a message `refuse`,
 - (b) submit a proposal for the contract, by sending a message `offer`.
3. **initiator**: if the received message is `refuse` then the algorithm ends.
4. **initiator**: if the received message is `offer` then decides one of:

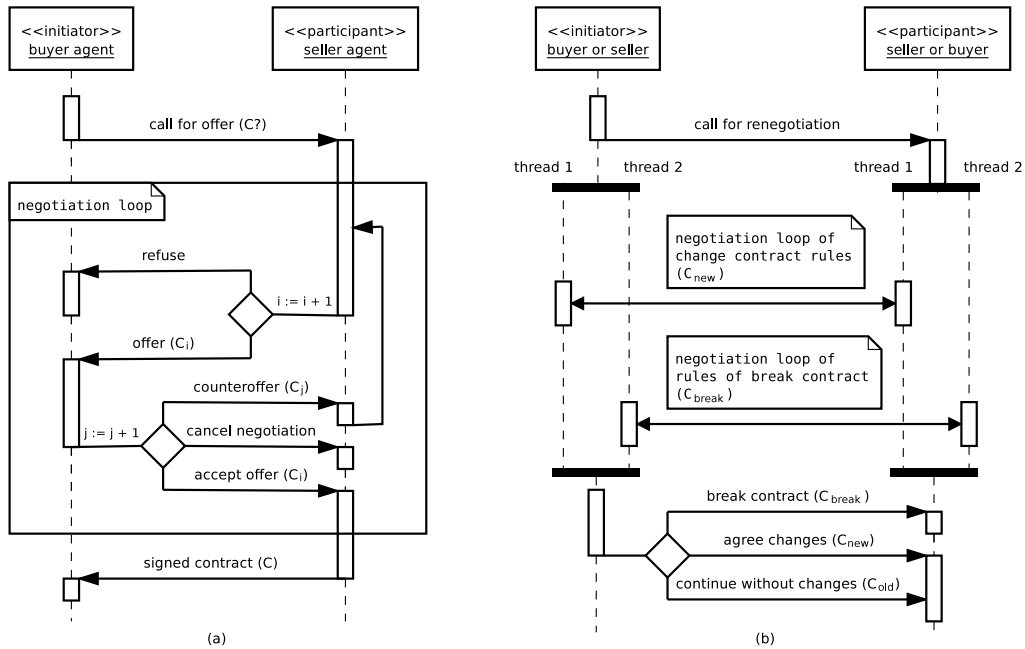


Fig. 2. Complex negotiations flow: (a) negotiations; (b) renegotiations.

- (a) accept offer with the contract terms received, by sending a message `accept offer`.
 - (b) make own contract proposal, by sending a message `counteroffer`.
 - (c) cancel negotiations, by sending a message `cancel negotiation`.
5. participant: if the received message is `cancel negotiation` then the algorithm ends.
 6. participant: if the received message is `counteroffer`, the negotiation loop returns to 2.
 7. participant: if the received message is `accept offer` then contract signs and the algorithm ends.

The messages `offer`, `counteroffer` and, optionally, `call for offer`, store the contents of the terms of the contract C received as proposed terms from the negotiation partner. The messages `refuse` and `cancel negotiation` mean breaking of negotiations. Message `accept offer` consents to the execution of the contract under the terms of the last message received, `offer`, and the message `signed contract` confirms the conclusion of the contract.

3.2. Renegotiation and execution phase of the contract

The execution phase of the contract begins after the conclusion phase. At this stage, both partners are obliged to fulfil their obligations to each other. If one party wants to change the obligation, they can start renegotiations.

The purpose of renegotiations is to negotiate changes to the contract arrangements. During renegotiation, changes to the terms of the contract (thread 1) and breach of the contract conditions (thread 2) are negotiated in parallel. The role of the initiator is played by the side which started renegotiations, and role of the participant is played by the other party. Renegotiations take place according to the following algorithm (shown in Fig. 2b):

1. initiator: starts renegotiations sending message `call for renegotiation`.
2. Both parties go to two threads and perform negotiations steps from 2 to 4. There is some change at point 4.
 - the message is not sent, but the thread quits and returns the negotiated terms of the contract value, which is ready to accept. It is assumed that the thread is successful. Other routes complete the algorithm and return an empty value.
3. initiator: writes to:
 - (a) C_{old} — conditions of current contract (as before renegotiations).
 - (b) C_{new} — conditions of new contract (if thread 1 was successfully finished).

(c) C_{break} — compensation conditions (if thread 2 was not successfully finished).

- Next, based on the utility function, the initiator decides on the most favourable conditions and sends them to the participant.

If the initiator chooses C_{old} then the execution phase is carried out under the terms of the old contract. If the initiator chooses C_{new} or C_{break} then the execution phase goes from the beginning under the new contract.

4. An example commerce simulation system

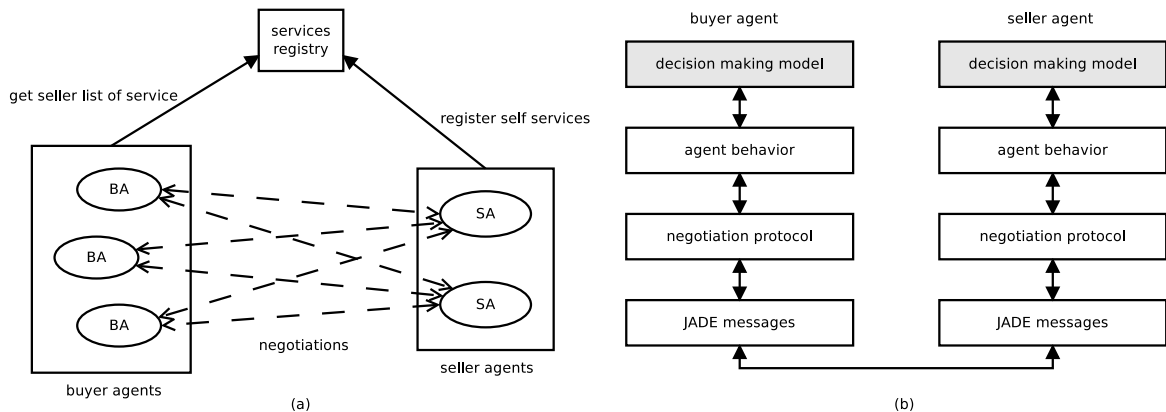


Fig. 3. Architecture of the example system: (a) multi-agent platform; (b) stack of the agent (gray - for expansion by developer).

The concept of complex negotiations has been verified in our multi-agent platform. The platform implements two types of agents (Fig. 3a):

- Seller Agent (SA).
- Buyer Agent (BA).

The Seller Agent is the manufacturer or vendor of the goods. During negotiations with customers, they are going to achieve the highest possible profit for themselves. If the Seller Agent wants to be found by the Buyer Agent, they must first of all register in the Services Registry (SR) a list of the types of goods g they are able to offer to the customer.

The Buyer Agent is responsible for the acquisition of goods. The Buyer Agent first gets the list of Seller Agents that are able to offer this kind of benefit, from the Service Registry. They then carry out their negotiations with the aim of signing the best possible contract C for themselves, for the delivery of the goods.

For each Seller Agent found, the Buyer Agent creates a separate thread. Each thread carries out negotiations with only one Seller Agent, but all communicate decisions to the main thread, which deals with the various Seller Agents and decides on further negotiations.

4.1. Implementation of agents

The implementation of agents was split into levels (Fig. 3b). The lowest level is the JADE agent platform. It was used as the agents runtime environment. The next level is the contract execution protocol (as presented above). This level describes the types of messages sent during negotiations, and during the conclusion and execution of the contract. The agent behaviour level manages the negotiations and renegotiations with the other party (Fig. 4). On this level, the algorithms for negotiation and renegotiation (described above) are performed, and the decision-making process of these algorithms refers to the upper layer. As has already been described, the algorithm branches to renegotiate the two threads. However, the Buyer Agent algorithm has been extended to a third topic. It carries out the conclusion phase of the contract, C with other vendors. In this way, the higher layer can decide on one of the following options:

- Interrupt current negotiations and retain the current contract conditions (C_{old}).

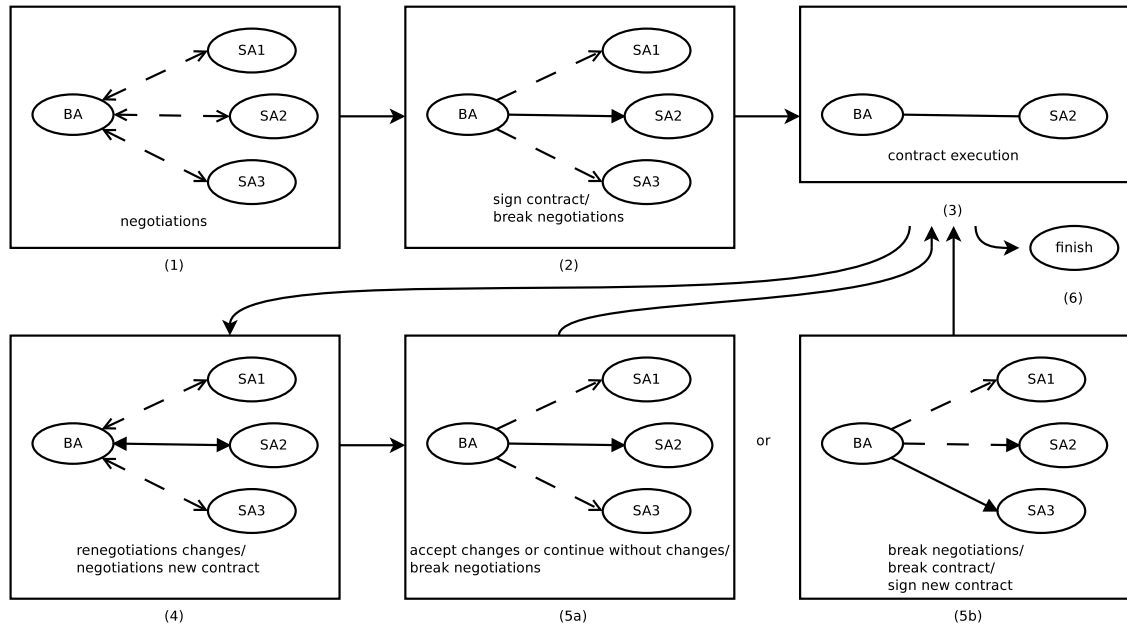


Fig. 4. Contract between buyer (BA) and sender (SAx) from the buyer point of view: (1) conclusion phase, (2) contract concluding, (3) execution phase, (4) contract renegotiation, (5a) continue with actual seller, (5b) cancel the contract and start cooperation with a new one, (6) finish the contract.

2. Cancel the contract (C_{break}).
3. Cancel the contract with the current contractor (C_{break}) and conclude new contract with a new contractor (C) — only the Buyer Agent.
4. Accept new contract conditions (C_{new}) — in this situation cancel old contract and sign new contract with the same contractor.

The highest layer is the decision-making model. This layer is responsible for negotiating strategy. It decides to start the negotiations, then it generates an offer or counteroffer and decides what type of response to send to the messages from the other party. To monitor communications and state agents we used [27].

4.2. Implementation of contract model

In this platform contract is expressed as follows:

$$C = (b, s, C_b, C_s) \quad C_b = (C_{b,1}, C_{b,2}, \dots, C_{b,n}) \quad C_s = (C_{s,1}, C_{s,2}, \dots, C_{s,m}) \quad (1)$$

$$C_{b,i}, C_{s,j} = (g, (F_1, F_2, \dots, F_n), q, t, e) \quad (2)$$

Where: C — contract, b — buyer, s — seller, C_b — buyer’s obligation, C_s — seller’s obligation, g — type of goods identifier, F_i — value of i-index goods feature, q — quantity of the good, t — time of delivery of goods (time to provide good service or time of service start), e — good validity (time when good loses value or time when service finishes).

By taking into account the characteristics of goods, we can assess their usefulness using a utility function f_u . The usability of the entire contract can be estimated on the basis of the utility function of the characteristics of liabilities and external conditions contained in the variable f_s :

$$f_u : C_{b,i}, C_{s,j} \rightarrow \mathbb{R} \quad f_{u,C}(C) = \sum_{C_s} f_u(C_{s,j}) - \sum_{C_b} f_u(C_{b,i}) \quad f_{u,S}(C) = f_{u,C}(C) + f_s \quad (3)$$

The counterdomain of the function is \mathbb{R} . Variable f_s includes everything that can affect the contract utility function, but which is not considered in this article (such as the contractors reputation, the cost of implementing additional changes to the renegotiation of the terms of the contract and other external conditions).

5. Example of negotiation process usage

This chapter presents a scenario of negotiations related to the purchase of goods in a store. This scenario was performed on the multi-agent platform described in section 4. The decision-making model is based on a utility function f_u , purchased goods C_s , and payment in kind C_b . The model uses a fixed penalty for changing the terms of the contract or breach of contract. In all cases, the good that is paid for the purchased good is money. For simplicity, the examples offered by vendors are in all cases fixed, but allow a change in the terms or the termination of the contract in return for appropriate compensation.

5.1. Scenario: buying computer parts in a computer shop

Lets consider an example: A store sells computer components with a warranty. We assumed that the shop is obliged to accept the return of goods up to two weeks from the date of purchase if the customer can prove that this product does not meet their requirements, and the goods show no signs of use. In fact, when the package is opened and the contents are used for a few days, it loses its value, but in this is omitted in this example, for the sake of simplicity. However, it does include the need to visit the shop, the waste of time, inconvenience and distress associated with the need to persuade the seller to agree to refund or exchange the goods.

Lets assume the customer wants to buy a hard drive. Using the notation proposed we conduct a simulation similar to the actual negotiation process:

$$C_s = (C_{HDD}) \qquad C_{HDD} = (g_{HDD}, (F_C, F_S), q_{HDD}, t_{HDD}, e_{HDD}) \qquad (4)$$

The type of good is $g_{HDD} = HDD$, capacity F_C and speed F_S are given in Table 1, and quantity $q_{HDD} = 1$, time of delivery $t_{HDD} = 0$, expiration time $e_{HDD} = \infty$.

Payment in all cases is:

$$C_b = (C_{MONEY}) \qquad C_{MONEY} = (g_{MONEY}, (), q_{MONEY}, t_{MONEY}, e_{MONEY}) \qquad (5)$$

Type of good is $g_{MONEY} = MONEY$, quantity q_{MONEY} are given in Table 1, time of delivery $t_{MONEY} = 0$, expiration time $e_{MONEY} = \infty$.

Lets assume that the client does not know how important speed is, so initially the disk utility function is dependent only on the capacity and may look like this:

$$f_u(HDD) = F_C \qquad f_u(MONEY) = q_{MONEY} \qquad (6)$$

$$f_{u,C} = f_u(HDD) - f_u(MONEY) \qquad (7)$$

As mentioned above, the replacement or return of goods involves some inconvenience and travel expenses, so the client will undertake this operation only when the value of the new offer in the renegotiation is sufficiently greater than the current contract. For this purpose, f_s is used, this being in the conclusion phase of the contract $f_s = 0$ and in the execution phase of the contract:

- for current contract $f_s = 0$,
- for other offers from this shop $f_s = -1$, where -0.6 means there is a psychological barrier against the need to persuade sellers to exchange goods, while -0.4 are the costs of travel and return from the store,
- for offers from other shops $f_s = -5$, where -4.4 means there is a psychological barrier against the need to persuade sellers to exchange goods, while -0.6 represents that total travel expenses between home and both shops.

The costs of travel to the shops accumulate after each exchange of goods. Stores offers is present on Table 1.

Thus, the client chooses A1 (see Fig. 4.1 to Fig. 4.3), which is a slow high-capacity disk (Table 1, negotiation). However, after three days he finds the speed to be important, so his utility function changes to the following:

$$f'_u(HDD) = 0.5F_C + 0.5F_S \qquad f'_u(MONEY) = q_{MONEY} \qquad (8)$$

As a result of changing preferences on the side of the client (see Fig. 4.4), if the purchasing decisions could be taken back the rank listings would look like Table 1 (renegotiation 1).

As you can now see, the most preferred choice is a fast high-capacity disk. It is offered by another store, which would involve the high cost of returning the goods to the original store.

Lets consider that, after a further three days, the client feels the need for a faster drive and the utility function changes to the following form:

$$f''_u(HDD) = 0.25F_C + 0.75F_S \qquad f''_u(MONEY) = q_{MONEY} \qquad (9)$$

As a result (Table 1, renegotiation 2), the value of offer B2 (very fast but the low-capacity SSD disk) has grown so much that it now outweighs the value of offer A1. The client decides to return the disk to a shop A and buy a new one in shop B (see Fig. 4.5b). The execution phase of the contract begins again from scratch.

After almost two weeks, the user finds the drive cannot hold all the data they want to store. Capacity has become more important in preference to function, so the importance of speed has decreased. Now the utility function is come back to (8):

$$f'''_u(HDD) = 0.5F_C + 0.5F_S \qquad f'''_u(MONEY) = q_{MONEY} \qquad (10)$$

Now the leader is offer B1, a fast high-capacity hard drive (Table 1, renegotiation 3). Because it is offered by the same shop, the value of $f_s = -1$. As a result, the customer decides to replace the goods with a new model (see Fig. 4.5a).

Table 1. Offers of sellers and utility function result.

Seller <i>s</i>	Offer <i>C</i>	HDD		MONEY <i>q</i> _{MONEY}	negotiation			renegotiation 1			renegotiation 2			renegotiation 3		
		<i>F</i> _C	<i>F</i> _S		<i>f</i> _{u,C}	<i>f</i> _s	<i>f</i> _{u,S}	<i>f</i> _{u,C}	<i>f</i> _s	<i>f</i> _{u,S}	<i>f</i> _{u,C}	<i>f</i> _s	<i>f</i> _{u,S}	<i>f</i> _{u,C}	<i>f</i> _s	<i>f</i> _{u,S}
A	A1	14	6	4	10	0	10	6	0	6	4	0	4	6	-5	1
B	B1	14	14	6	8	0	8	8	-5	3	8	-5	3	8	-1	7
B	B2	2	22	7	-5	0	-5	5	-5	0	10	-5	5	5	0	5

5.2. Evaluation of the results

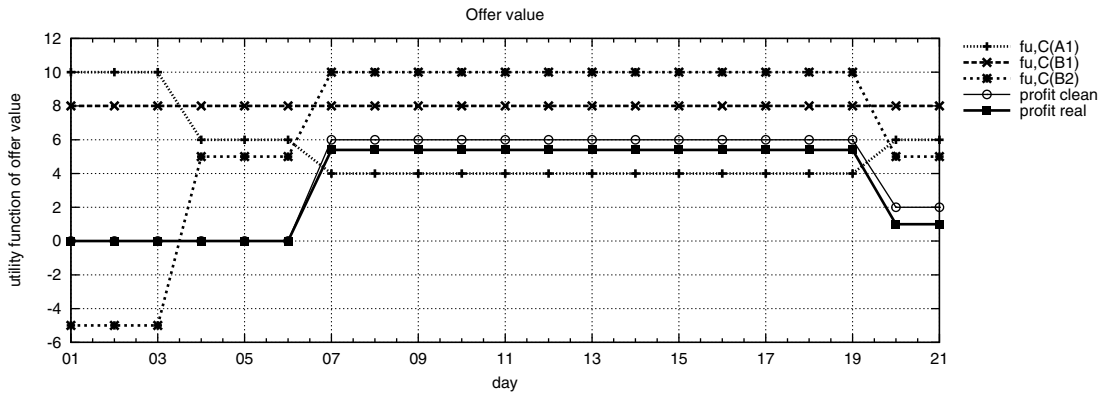


Fig. 5. Value of offers A1, B1 i B2 in time and profit as a result of exchange of goods (profit clean — only profit, profit real — profit with the culmination of the cost of replacing).

The graph shown in Fig. 5 shows the results of the simulation. Lines A1, B1 and B2 represent the value of the offer at the time. The line clean profit represents the difference between the actual conditions of the contract C_{old} and offer value A1 that is responsible for the contract that is concluded as result of negotiation. The line real profit includes the accumulated cost of replacing the product, which was used as a measure of the benefits of renegotiation.

In this scenario, renegotiations gave the client the opportunity for such an exchange at the time it was most needed. As can be seen in the graph, the first change in preferences happened on the fourth day. The difference

between the old choice of utilities and the then best offer was lower than the value of f_s . Therefore, the Buyer Agent decided not to exchange. The decision to exchange happened on the seventh day, when the gap had growth and became higher than lowering the impact of f_s . As a result, in the following days the Buyer Agent had a more valuable contract than they had negotiated at the beginning.

After each renegotiation where a change of contract took place, the cost of changes accumulated. In this example, we have assumed that the accumulation of these costs was simply the sum of f_s , which was responsible for the cost of getting to the shops. On the graph it is visible in the form of the difference between profit clean and real profit.

6. Summary

This paper presents a new approach to negotiations. This approach extends the process across the lifetime of the contract. It allows changes in the terms of the contract, as negotiated at the time of its implementation.

The example demonstrates a simulation of the process of using this approach. The profit of the renegotiation and the costs resulting from changes to the terms of the contract are presented in the example. These costs accumulate after every change, so if changes are performed too often during the execution of the contract, the costs can exceed the value of the profit. This is why it is important to choose an appropriate strategy and relevant parameters. In some cases, the cost of changes to the contract terms can be very high, and it is better to predict future changes in preferences and include them in the negotiations. In other cases, the cost of renegotiation can be almost zero and complex negotiations should have the greatest success.

It is necessary to undertake research on renegotiation strategies. Renegotiations are not normal negotiations. In renegotiation it has to be considered that the contract has been partially realized. Different kind of changes may generate different costs.

Further research into complex negotiation will include its use in supply chain management [28]. It enables a dynamic response to the changing needs and capabilities of each cell in the chain, including failures, downtime in production and managing unused resources (the use of goods which have been discontinued as a result of breach of contract). There are plans to study the merger of the same cells in dynamic coalitions where some agents may dynamically attach to and detach from the coalition.

All of these issues can be applied in production control systems, forwarding or trade.

Acknowledgements

The research leading to these results has received funding from the research project No. O ROB 0008 01 “Advanced IT techniques supporting data processing in criminal analysis”, funded by the Polish National Centre for Research and Development., and from dean grant no. 15.11.120.266

References

- [1] G. Wang, T. Wong, X. Wang, An ontology based approach to organize multi-agent assisted supply chain negotiations, *Computers & Industrial Engineering* (0) (2012) –. doi:10.1016/j.cie.2012.06.018.
URL <http://www.sciencedirect.com/science/article/pii/S0360835212001775>
- [2] P. Kolarz, J. Marszałek, J. Koźlak, M. Żabińska, Multi-agent environment for modelling and analysing market strategies, *Procedia Computer Science* 9 (2012) 1533–1542, proceedings of the International Conference on Computational Science, 2012.
- [3] R. Smith, The contract net protocol: High-level communication and control in a distributed problem solver, *IEEE Transactions on computers* 29 (12) (1980) 1104–1113.
- [4] FIPA, FIPA Contract Net Interaction Protocol Specification, FIPA (2001).
URL <http://www.fipa.org/specs/fipa00029/>
- [5] M. Calisti, An agent-based approach for coordinated multi-provider service provisioning, *Whitestein series in software agent technologies*, Birkhauser, Basel, Boston, Berlin, 2003.
URL <http://opac.inria.fr/record=b1119048>
- [6] C.-C. Huang, W.-Y. Liang, Y.-H. Lai, Y.-C. Lin, The agent-based negotiation process for b2c e-commerce, *Expert Systems with Applications* 37 (1) (2010) 348 – 359. doi:10.1016/j.eswa.2009.05.065.
URL <http://www.sciencedirect.com/science/article/pii/S0957417409004990>

- [7] R. Y. Lau, Towards a web services and intelligent agents-based negotiation system for b2b ecommerce, *Electronic Commerce Research and Applications* 6 (3) (2007) 260 – 273. doi:10.1016/j.elerap.2006.06.007.
URL <http://www.sciencedirect.com/science/article/pii/S1567422306000433>
- [8] G. Wang, T. Wong, C. Yu, A computational model for multi-agent e-commerce negotiations with adaptive negotiation behaviors, *Journal of Computational Science* (0) (2011) –. doi:10.1016/j.jocs.2011.10.003.
URL <http://www.sciencedirect.com/science/article/pii/S1877750311000895>
- [9] D.-Y. Kim, P. Humblet, M. Eyuboglu, L. Brown, J. Forney, G.D., S. Mehrabanzad, V.92: the last dial-up modem?, *Communications, IEEE Transactions on* 52 (1) (2004) 54 – 61. doi:10.1109/TCOMM.2003.822168.
- [10] H. Song, K. M. Lee, Adaptive rate control algorithms for low bit rate video under networks supporting bandwidth renegotiation, *Signal Processing: Image Communication* 17 (10) (2002) 759 – 779. doi:10.1016/S0923-5965(02)00007-3.
URL <http://www.sciencedirect.com/science/article/pii/S0923596502000073>
- [11] C.-T. Chang, Multi-choice goal programming with utility functions, *European Journal of Operational Research* 215 (2) (2011) 439 – 445. doi:10.1016/j.ejor.2011.06.041.
URL <http://www.sciencedirect.com/science/article/pii/S0377221711005704>
- [12] Y. M. Chen, P.-N. Huang, Agent-based bilateral multi-issue negotiation scheme for e-market transactions, *Applied Soft Computing* 9 (3) (2009) 1057 – 1067. doi:10.1016/j.asoc.2009.02.008.
URL <http://www.sciencedirect.com/science/article/pii/S1568494609000295>
- [13] R. A. Carbonneau, G. E. Kersten, R. M. Vahidov, Pairwise issue modeling for negotiation counteroffer prediction using neural networks, *Decision Support Systems* 50 (2) (2011) 449 – 459. doi:10.1016/j.dss.2010.11.002.
URL <http://www.sciencedirect.com/science/article/pii/S0167923610001831>
- [14] V. Sanchez-Anguix, S. Valero, V. Julin, V. Botti, A. Garca-Fornes, Evolutionary-aided negotiation model for bilateral bargaining in ambient intelligence domains with complex utility functions, *Information Systems* 222 (0) (2013) 25 – 46, *Including Special Section on New Trends in Ambient Intelligence and Bio-inspired Systems*; *cc: title*. doi:10.1016/j.ins.2010.11.018.
URL <http://www.sciencedirect.com/science/article/pii/S0020025510005633>
- [15] H. Kebriaei, A. Rahimi-Kian, V. J. Majd, An agent-based system for bilateral contracts of energy, *Expert Systems with Applications* 38 (9) (2011) 11369 – 11376. doi:10.1016/j.eswa.2011.03.005.
URL <http://www.sciencedirect.com/science/article/pii/S0957417411004131>
- [16] B. An, K. Sim, L. Tang, C. Miao, Z. Shen, D. Cheng, Negotiation agents decision making using markov chains, in: T. Ito, H. Hattori, M. Zhang, T. Matsuo (Eds.), *Rational, Robust, and Secure Negotiations in Multi-Agent Systems*, Vol. 89 of *Studies in Computational Intelligence*, Springer Berlin Heidelberg, 2008, pp. 3–23. doi:10.1007/978-3-540-76282-9_1.
URL http://dx.doi.org/10.1007/978-3-540-76282-9_1
- [17] H. C. Lau, W. S. Lim, Multi-agent coalition via autonomous price negotiation in a real-time web environment, in: *Intelligent Agent Technology*, 2003. IAT 2003. IEEE/WIC International Conference on, 2003, pp. 580 – 583. doi:10.1109/IAT.2003.1241149.
- [18] T. Wanyama, B. H. Far, Negotiation coalitions in group-choice multi-agent systems, in: *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, AAMAS '06*, ACM, New York, NY, USA, 2006, pp. 408–410. doi:10.1145/1160633.1160704.
URL <http://doi.acm.org/10.1145/1160633.1160704>
- [19] B. Martinovski, Emotion in negotiation, in: D. M. Kilgour, C. Eden (Eds.), *Handbook of Group Decision and Negotiation*, Vol. 4 of *Advances in Group Decision and Negotiation*, Springer Netherlands, 2010, pp. 65–86. doi:10.1007/978-90-481-9097-3_5.
URL http://dx.doi.org/10.1007/978-90-481-9097-3_5
- [20] A. Monteserin, A. Amandi, Argumentationbased negotiation planning for autonomous agents, *Decision Support Systems* 51 (3) (2011) 532 – 548. doi:10.1016/j.dss.2011.02.016.
URL <http://www.sciencedirect.com/science/article/pii/S016792361100087X>
- [21] T. Coltman, K. Bru, N. Perm-Ajcharyawong, T. M. Devinney, G. R. Benito, Supply chain contract evolution, *European Management Journal* 27 (6) (2009) 388 – 401. doi:10.1016/j.emj.2008.11.005.
URL <http://www.sciencedirect.com/science/article/pii/S0263237308001333>
- [22] M. De Marco, V. Trecordi, Bandwidth re-negotiation in atm networks for high-speed computer communications, in: *Global Telecommunications Conference, 1995. GLOBECOM '95., IEEE*, Vol. 1, 1995, pp. 393–398 vol.1. doi:10.1109/GLOCOM.1995.501957.
- [23] D. Denteneer, V. Pronk, J. Griffiths, L. Cuthbert, Impact of the resource needed for renegotiating atm rates, *Computer Networks* 34 (1) (2000) 211 – 225. doi:10.1016/S1389-1286(00)00107-9.
URL <http://www.sciencedirect.com/science/article/pii/S1389128600001079>
- [24] D. Batre, F. Brazier, K. Clark, M. Oey, A. Papaspyrou, O. Waldrich, P. Wieder, W. Ziegler, A proposal for ws-agreement negotiation, in: *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*, 2010, pp. 233–241. doi:10.1109/GRID.2010.5697976.
- [25] K. Cetnarowicz, M-agent architecture based method of development of multiagent systems, in: *Proc. of the 8th Joint EPS-APS International Conference on Physics Computing, ACC Cyfronet*, 1996.
- [26] K. Cetnarowicz, T. Dyduch, J. Koźlak, M. Zabińska, P. Błaszczyk, M. Niedźwiecki, K. Rzecki, L. Belava, G. Wachocki, P. Bech, J. Dziedzic, M. Ptaszek, SOA Infrastructure Tools - Concepts and Methods, *Poznań University of Economics Press*, 2010, Ch. SOA-Based Multi-Server Agent System — Application for Integrated Rescue Action, pp. 471–488.
- [27] M. Niedźwiecki, K. Rzecki, P. Błaszczyk, Asynchronous agent system for monitoring communication and system states based on the soa paradigm, in: *European Council for Modeling and Simulation, ECMS 2011 Proceedings* edited by: T. Burczynski, J. Kolodziej, A. Byrski, M. Carvalho, 2011, pp. 55–60. doi:10.7148/2011-0055-0060.
- [28] E. Nawarecki, J. Koźlak, Building multi-agent models applied to supply chain management, *Control and Cybernetics* 39 (1) (2010) 149–176.