

Multi-agent Contracting and Reconfiguration in Competitive Environments using Acquaintance Models

Jiří Bíba, Jiří Vokřínek

Gerstner Laboratory, Dept. of Cybernetics, Czech Technical University, Technická 2, 166 27 Praha 6, Czech Republic

biba@labe.felk.cvut.cz, vokrinek@labe.felk.cvut.cz

Abstract. *Cooperation of agents in competitive environments is more complicated than in collaborative environments. Both replanning and reconfiguration play a crucial role in cooperation, and introduce a means for implementing a system flexibility. The concepts of commitments, decommitments with penalties and subcontracting may facilitate effective reconfiguration and replanning. Agents in competitive environments are fully autonomous and self-interested. Therefore the setting of penalties and profit computation cannot be provided centrally. Both the costs and the gain differ from agent to agent with respect to contracts already agreed and resources load. This paper proposes an acquaintance model for contracting in competitive environments and introduces possibilities of reconfiguring in competitive environments as a means of decommitment optimization with respect to resources load and profit maximization. The presented algorithm for contract price setting does not use any centralized knowledge and provides results corresponding to a realistic environment. A simple customer-provider scenario proves this algorithm in competitive contracting.*

Keywords

multi-agent systems, competitive environments, reconfiguration, decommitment, contract price, penalty

1. Introduction

A concept of multi-agent systems (MAS) is a widely used paradigm for modelling, planning and control of various processes. Generally, it uses distributed negotiation techniques for achieving particular goals. Besides standard centralized planning and optimization mechanisms, MAS supports local replanning with minimal needed changes of the entire plan. There are several MAS implementations for production planning – e.g. [16] and for cooperation across supply chains [18, 20, 13]. Modern business speeds up research in the domain of Virtual Organizations [10] that trans-

form supply chains into dynamic cooperative networks¹. Cooperation in such an environment is based on distributed negotiation among individual partners that leads to satisfaction of individual or common goals. In case of internal cooperation (within an enterprise), the goal is to maximize the overall profit of the whole enterprise. Changing the scope to external cooperation (across a supply chain), the behaviour of the parties involved is more self-interested, as their goal is to maximize their own profits. Standard negotiation protocols and techniques used in MAS do not follow this course, so new negotiation principles for such an environment have to be investigated.

2. Competitive and Collaborative Environments

Let us introduce a difference between *collaborative* and *competitive* multi-agent environments [1]. By a *collaborative* multi-agent environment we understand an agent community where the agents usually share a common goal that they try to achieve cooperatively. In other cases the agents may have different goals, but their primary motivation is to maximize their social welfare – the total sum of all the individual utilities (profits) of the collaborative agents. Conversely, by a *competitive* multi-agent environment we understand an agent community where the primary motivation of the agents is to maximize their individual utilities, no matter what the social welfare of the community is (agents are so called *self-interested*). The agents establish cooperation in the process of achieving a common goal only if it maximizes their individual utilities.

As mentioned above, cooperation across a supply chain differs substantially from the cooperation within an enterprise. We distinguish collaborative and competitive environments and inspect different aspects of cooperation in these cases. In our work we focus on reconfiguring and replanning as the crucial element in successful cooperation among agents. This paper attempts to analyze ways of adjusting contracts in a real-world setting. Proper algorithms for con-

¹Cooperation with the other partners in a Virtual Organization allows the enterprise to react to incoming business opportunities that could not be covered by the enterprise alone.

tracting need to be developed as they underlie the cooperation in a competitive environment.

3. Commitments and Decommittments

The concept of cooperative problem solving by means of social commitments was introduced by Wooldridge and Jennings in 1999. Dropping a social commitment (decommitment) was either rational and beneficial for all the participants, or did not occur at all. Although the authors do not restrict the commitment description provided here to collaborative environments, the agents turn out to be social-welfare maximizers rather than competitors. However, in competitive environments an agent tends to drop its commitments if this maximizes its individual utility, no matter how it may consequently harm the others. If we want self-interested agents either to fulfil their commitments or to provide compensations for the harm to others in the case of decommitment (i.e. if we want agents to act responsibly), the agents have to commit themselves in this sense as well.

For contracting in collaborative environments there is usually no need for any explicit metrics of the individual utility or the social welfare gained². By contrast, in competitive environments an explicit expression of utility is desirable. It facilitates the implementation of rewards and penalties as utilities that the agents gain or lose. The concept of such an explicit utility evaluation is then a part of commitments – an agent providing a service (the *contractee*) commits not only to perform appropriate actions (in order to gain the utility promised – this is the agent’s motivation), but to provide compensation if fails (e.g. a compensation of the profit lost to the other party). Simultaneously, the other party (the *contractor*) commits not only to pay for services provided by the first party, but also to provide a compensation if it decommits from the contract (the first party suffers loss of profit and is paid e.g. the opportunity cost).

3.1. Levelled Commitment Contracts

The most complete approach to commitments in a competitive environment has been presented by Sandholm and Lesser [19] as *levelled commitments*. Levelled commitments include an explicit utility evaluation in the form of a *contract price* and *penalties*. Levelled commitments facilitate decommitments that were not acceptable for *full commitments* commonly used. A full commitment is defined by a contract obligation as an n-tuple (Γ, ρ) , where Γ introduces a description of what each of the two parties (the contractor and the contractee) has to perform (handling tasks, contributing goods, lending resources, etc.) and ρ introduces the contract price that the contractor has to pay to the contractee.

²For example, the number of goals successfully achieved suffices as an evaluation of the utility gained; algorithms for contracting in collaborative environments often guarantee maximization of social welfare.

Neither of the agents may drop the commitment, under any circumstance till it is brought to a good end. In contrast, *levelled commitments* are defined as an n-tuple (Γ, ρ, a, b) , where the extending parameters a and b introduce penalties to be paid when decommitments occur.

Levelled commitments are based on non-cooperative game theory. A negotiation process consists of two parts – (i) the *contracting game*, when the agents agree on a contract and (ii) the *decommitting game*, when they decide whether or not to decommit. Various events may occur (resources failing or becoming available, outside offers, etc.) that change the value of the contract independently for any of the two agents so that keeping the commitment does not need to be desirable for one or for both of the agents. Both the decommitment decision and the setting of the contract (ρ, a and b) are based on knowledge of the *ex ante* probability density functions (p.d.f.) of receiving the best outside offers. The p.d.f. are assumed to be common knowledge between the contractor and the contractee.

Levelled commitments have several limiting assumptions that facilitate equilibrium calculations of the contract settings, but make the use of levelled commitments more difficult in domains where such assumptions may be neither possible nor even desirable (e.g. logistics, production planning, etc.). The most significant assumptions are: (i) an agent does not want to be involved in more than one contract at a time, (ii) all the contracts available have the same description Γ (the only concern is the contract price) and (iii) the p.d.f. of receiving the best outside offers are common knowledge between the agents. The most limiting assumption is (iii) [8].

Moreover, the concept of levelled commitments also does not state explicitly whether the contract price under consideration introduces only a profit or whether it considers costs on performing Γ . It rather seems that ρ introduces the total price of the contract, set only on the basis of p.d.f. (there is no distinction between costs and the expected profit that both are comprised in the real-world contract price). Fixed costs are seemingly also not taken into account, as the price of a *null deal* (i.e. the agents do not agree on a contract) is assumed to be equal to the average best outside offer. Thus, the agent does not lose anything, but only does not get what it might have got if the best offer had come.

3.2. Contract Setting

An extension of levelled commitment contracts has been introduced by Excelente-Toledo *et al.* in [8]. They provide both an algorithm for calculating the contract setting and an algorithm for considering a decommitment. Unfortunately, the assumptions considered (e.g. omitting the fixed costs) need not always be acceptable. Thus, algorithms for

contract setting in the competitive environments need to be developed.

The price of a contract in the real world covers at least the following three items: (i) *variable costs* which depend on contract size, feasibility issues, etc. (i.e. specific conditions related to a particular contract), (ii) *fixed costs* which are not related to a particular contract, but are related to the overall business and need to be covered (e.g. the rent for an office, payment for energy, employees' wages, etc.) and (iii) *intended profit* from the contract (e.g. a profit for the enterprise owner). A penalty in the real world seeks to cover at least a portion of the fixed costs and also the *profit lost*. While the calculation of variable and fixed costs or of profit lost, is rather pragmatic, the setting of the intended profit is rather strategic or even speculative, and depends on many considerations (e.g. experience with the second party, various social relations, the profit eagerness of the first party or "good manners"). Overall, the setting of a contract price and a penalty may predetermine the acceptability of such a bid for the customer, i.e. the fruitfulness of the contact.

Let us propose an algorithm for setting a contract price. The scenario is as follows: there are two actors – a customer and a service provider. The customer proposes contracts of different sizes and calls for bids. The service provider calculates the bids and proposes the prices to the customer. Let the coefficient of variable costs with respect to contract size be common for all agents, and let the customer's private preference be to accept bids with a margin up to e.g. 10% of the variable costs³. The service provider does not try to do more than cover all the costs (variable and fixed). It calculates the variable costs and projects the actual fixed costs to be covered to the margin⁴. Let the margin be finally limited in three different ways: (i) *simple limitation*, (ii) *learned safe limitation* and (iii) *learned speculative limitation*:

simple limitation – the margin is not limited until it attacks a certain bound with respect to the variable costs – let us say 50% of the variable costs

learned safe limitation – the margin is limited either to an average value of the previously accepted margins or to a half of a minimum previously rejected margin

learned speculative margin – the margin is limited to mid way between the maximal accepted (lower boundary) and minimal rejected (upper boundary) margin from the past; until both boundaries are obtained the margins are either increased by 50% or decreased by 20%.

Obviously, the first approach does not guarantee the provider a profit in the long-term. The other two approaches do, because the provider learns from the past (provided the

³This is inspired by the real-world contracting, where a customer is willing to accept a price only up to a certain limit.

⁴The fixed costs are constant for a time unit, but are generally accumulated or reduced based on whether the provider's business was successful in the past.

total turnover can cover the fixed costs). The latter limitation then promises the maximum possible profit (experiments are described in the section 4).

3.3. Negotiation and Acquaintance Model for Competitive Contracting

The process of contracting and contract maintenance consist of two fundamental processes: (i) *negotiation* and (ii) *deliberation*. Negotiation is a process of exchanging messages among the agents in accordance with interaction protocols agreed in advance. Deliberation is a complex process of knowledge maintenance, evaluation of different decision alternatives, and choice of the appropriate one (decision making) with respect to knowledge of the actual situation in the surrounding environment, decision strategies and individual goals.

Many interaction protocols implementing various negotiation methodologies have been developed. A well known and very popular methodology is the a Contract-Net Protocol, which implements a CNP auction that has been standardized by FIPA [9]. However, this interaction protocol (and most of the widely-used related protocols, some of them also implemented by FIPA), was developed rather for collaborative environments [5]. The most interesting multi-stage negotiation protocol for flexible negotiation in competitive environments was presented by Bergenti *et al.* in [2]. This protocol allows the agents (the *initiator* and the *responder*) to propose and to counter-propose till they reach an agreement, or until one of them decides to withdraw from the negotiation. If agreement is reached, any of the parties may decommit and the contract becomes void. If the contract is successfully completed, the responder informs the initiator about it.

One of the means used for decision making involves *acquaintance models*. An *acquaintance model* is a computational model of agents' mutual awareness, stored in the interaction wrapper of each of the agents. The acquaintance model is a collection of the agent's social knowledge [14] available from previous interactions or provided by independent monitoring mechanisms. There are various implementations of acquaintance models, e.g. the *tri-base* (3bA) acquaintance model [17], the *twin-based model* [4] or the acquaintance model in ARCHON [21].

The contracting algorithm proposed in section 3.2 uses quite a simple acquaintance model that enables only tracking the history of contracts for each customer that the provider interacts with, and the decision making process is rather myopic. Moreover, in a competitive environment more considerations need to be employed, because it is necessary to reflect e.g. contract profitability, partners' reputations, actual contract commitments, availability of resources, long-term strategies, etc. Therefore, we propose an acquaintance model consisting of three modules:

profitability module – maintains both the own economy model and the economy models of the partners; the module determines the profitability of incoming contract offers from various points of view – e.g. their contribution to covering fixed costs, attainable profit, increase/decrease of own reputation, a strategic measure taking into account long-term cooperation with the particular partner, etc.

reputation module – maintains track of former interactions with partners. These are used for building reputation models for all partners as well as an estimate of one's own reputation with the other partners; the module determines the safety/risk of concluding a contract (both incoming and outgoing opportunities) with a particular partner, and advises on decisions or adjustments concerning the contract – e.g. whether or not conclude the contract, under which conditions to conclude it with respect to possible delays (setting a reserve in service delivery time), service quality or total failure (setting penalties), payment delays (setting a reserve in the contract price as opportunity cost insurance), etc.

commitments–resources module – maintains both a record of the actual contracts (i.e. both the commitments of the agent to the partners and the commitments of partners to the agent) and user's own resource availability schedule bound with the contracts; the module supports scheduling of incoming contracts, provides information about resource availability and facilitates computation of the feasibility of the contract, simulation of overbooking of resources with a risk evaluation (i.e. level of penalties for delays or decommitments in the case of unavailability of resources), simulation of resource relaxation based on outsourcing, etc.

As the contracting process and contract maintenance in a competitive environment are based on the agent's local knowledge only, the acquaintance model is built iteratively based on the experience of the agent during interactions with the partners (other agents). Thus, model building is an inseparable part of the decision making processes. The acquired knowledge is then used by an inference machine based on a rule system that either provides advice to a human operator (manager) concerning the incoming and ongoing contracts or carries out decisions by means of pre-defined rules (i.e. contract acceptance or rejection, decommitments or outsourcing, etc.). Administrative activities like updating the company ERM/CRP system or payments to partners are then carried out automatically.

3.4. Reconfiguration in Competitive Environments

Although there are some domain-dependent application implementations of reconfiguration in multi-agent systems (MAS) - e.g. [6, 11, 3], the first deep study of a reconfiguration and its formalization was published only recently published by Dunin-Keplics in [7]. The multi-agent environment has been assumed to be collaborative. Thus, if a failure occurred, all the agents involved in achieving their common goal did their best to establish a recovery, in order to complete their task successfully (a reconfiguration occurred). This behaviour due to by their persistent collective intention to achieve their common goal in accordance to the definition of a joint persistent goal [12].

However, in competitive environments the collective intention does not need to be kept unconditionally by all the agents – any agent may decommit from the actual contract (e.g. on account of a more profitable third-party contract offer). Decommitment as a concept was not taken into account in the above-mentioned research, while according to [22] it may become a means for optimizing the agents' individual profits.

While the implementation of a reconfiguration in collaborative environments was facilitated by the agents' primary motivation (i.e. maximization of their social welfare), in competitive environments it is even more difficult. In collaborative environment, decommitment or replanning is driven by a common goal. It is obvious that both partners to the 'contract' have the same motivation to keep it or change it. In a competitive environment, the agents need to be motivated not only to agree on a contract and to keep their commitments, but also to perform a reconfiguration, if necessary. A self-interested agent will be reluctant to take on further obligations if they are not rewarded, or if they even entailed lost of profit.

One of the available motivations is to use reconfiguration as an alternative to decommitment, if keeping the current contract does not contribute to the maximization of the agent's individual utility (e.g. if a more profitable offer has appeared). For example, the agent may find a subcontract that can be reimbursed from the reward promised by its contractor and may decide to take advantage of both the current contract and the new contract without decommitting. Of course, the idea of decommitment may have resulted from of unanticipated events, e.g. lack of resources, an unexpected delay that may prevent the contract deadline being met, etc. Thus, reconfiguration may appear to be useful for optimizing the resource load and also for maximizing the profit by optimizing any possible or necessary decommitments. In a collaborative environment reconfiguration is usually driven by individual tasks (it is invoked top-down by decomposition) rather than by the intention of the contracted agents. Table 1 shows an overview of various environment properties.

collaborative environments:

maximized criteria	social welfare
commitments	full
decommitments	common-goal driven
reconfiguration	contract based

competitive environments:

maximized criteria	individual utility
commitments	full, levelled
decommitments	individual-utility driven
reconfiguration	contracted-agent based

Tab. 1. Overview of various environment properties

Let us provide an example of a reconfiguration and its potential. *Customer 1* grants a contract *Contract 1* to a group of agents – coalition [15] – coordinated by *Provider Leader*. While the contract is being executed, a coalition member *Provider Traitor* receives a proposal for a more profitable contract *Contract 2* and decides to participate in it. However, its resources are not sufficient for both contracts, and therefore *Provider Traitor* considers a decommitment from *Contract 1*. It may still save the decommitment penalty if it finds a subcontractee *Provider Subcontractee* that takes on its obligations.

If *Provider Traitor* succeeds and subcontracting is less costly than fulfilling *Contract 1* by *Provider Traitor*, it may benefit on both contracts. The maximal rational cost of subcontracting is the sum of the profit on fulfilling *Contract 1* and the decommitment penalty for *Contract 1*. If *Provider Traitor* does not succeed in subcontracting, it decommits from *Contract 1*. In this case, although *Provider Leader* collects the decommitment penalty from *Provider Traitor*, it tends to find subcontracting on its own in order to avoid paying a penalty to *Customer 1*. If this succeeds, the coalition leader may keep the penalty as a reward and perhaps also the difference in the prices, provided the subcontracting is cheaper than *Provider Traitor*'s services. If it does not succeed, it pays the collected penalty to *Customer 1*.

There may be other reconfiguration scenarios – e.g. *Provider Leader* may receive a more profitable bid from *Customer 2*, or it may find a *Provider Substitutor* that may be cheaper than one of the current coalition members. However, reconfiguring and taking advantage of such opportunities is even more difficult than in the above-mentioned example.

In any case, reconfiguration strongly depends on the contract setting and vice versa. This needs to be reflected in the contracting algorithm. Extending the basic algorithm proposed in section 3.2, in this sense, requires taking into account, e.g. long-term business strategies, reputation, etc., as the basic version is still rather guesswork than a deliberative calculation. The implementation of a more sophisticated contracting algorithm using the proposed acquaintance model and thus capable of reconfiguration and also handling

e.g. customer's adaptive behaviour is currently under research.

4. Experiments

The provided experiments show the properties of our algorithm (in three variants), as proposed in section 3.2. Our model implementation defined two types of agents – the *customer* and the *provider*. The sizes of the contracts proposed by the customers were arbitrary, within a bounded interval and varied in number from none to several per day. The customer's margin boundary was set to 12%. The variable-costs coefficient and the provider's daily fixed costs were set to fixed values. For each algorithm variant 50 runs of the simulation were performed and the results were averaged⁵. Let us describe the results in detail. Fig. 1 introduces a comparison of the margin setting for all three algorithm variants with respect to the customer's margin boundary.

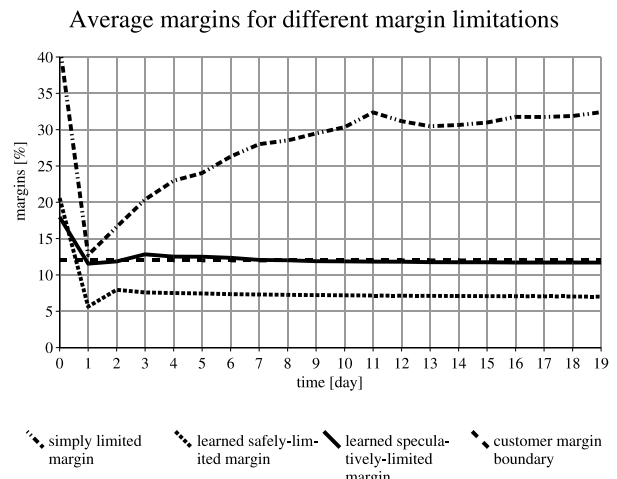


Fig. 1. Comparison of all the proposed limitations: average margins and customer's margin boundary

The first variant of the algorithm implements simple limitation of margins. As the agent does not take into account past experience, and its only concern is to cover its current fixed costs, the success ratio (the ratio of the number of successful contracts to the number of all contracts) decreases together with the growth of the debts reflected in the margin set (see figure 2). Although a contract sometimes arrives that is big enough for the fixed costs to be dissolved in the overall contract price and the bid is accepted (the customer's margin boundary is not attacked, and thus the fixed costs are covered), this happens only seldom, and mostly the agent has to cover its expenses from its reserve. This is the most obvious drawback of this algorithm variant – once the agent gets into debts, there is a little probability that it will get a chance to pay them back.

⁵Due to both the randomness of the contract sizes and the absence of contracting history, the mean square deviation at the first day was cca 0.24. However, it tended to decrease to zero quickly during the simulation.

The second variant implements the learned safe limitation of margins. In the first run, the agent sets the margin so that it covers its current fixed costs. If the bid is rejected the margin is reduced. This is done until a bid is accepted. Then the agent sets the margins to correspond to average accepted-margin value. The margin set in a particular contract does not need to cover all the daily fixed costs. However, if several contract proposals arrive in one day, they can cover the fixed costs when added together or even make a reserve for the future (the provider benefits from turnover). Such a strategy is safe, as it guarantees the acceptance of all bids (the margins are set to a certain value below the customers' margin boundaries) and the provider may even be better off over a period of time. Obviously, the provider does not get as much as it might have got if it had chosen a less safe strategy. Moreover, there a situation may occur in which the benefit from the turnover does not suffice to cover the agent's fixed costs and the agent may end up with debts. On the other hand, the process of running into debt may be slower than in the first algorithm variant.

The third variant implements the learned speculative limitation of margins. At the beginning of the simulation the provider speculates and tries to learn each customers' margin boundary. Thus, it is less successful at the beginning. On the other hand, once it approaches the best possible and most reasonable margin, it begins to benefit from the turnover. The learned margin guarantees acceptance of all bids and also the maximum pay-off from the particular customer. It may also run into debt, if the customers' margins are too low and the maximum benefit from the turnover does not cover the provider's fixed costs. On the other hand, the provider cannot defend himself against this situation and it is obvious that it will run into debt at the minimum possible speed.

Average success ratios for different margin limitations

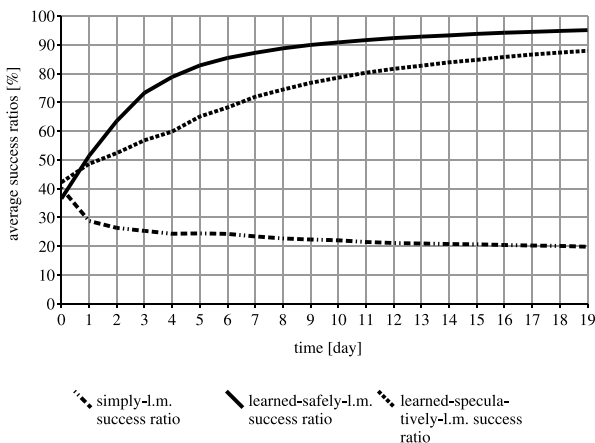


Fig. 2. Comparison of all the proposed limitations: average success ratios

Fig. 2 and Fig. 3 introduce a comparison among all three algorithm variants. While in the first variant the provider runs into debt, in the second and the third variant it begins

Account ratios for different margin limitations

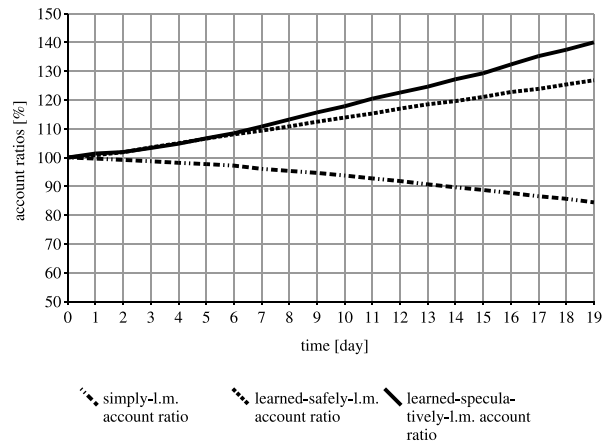


Fig. 3. Comparison of all the proposed limitations: account ratios

to be better off. Although the average success ratio of the third variant grows more slowly at the beginning than the success ratio of the second variant (due to speculation at the beginning), the account balance shows that the profit gained was worth incurring a loss at the beginning. Let it be noted that the success ratio was computed with respect to whole the simulation duration, i.e. the third-variant success ratio would converge to the second-variant success ratio, which converges to 100% in an infinite time horizon⁶.

5. Conclusion

This paper focuses on cooperation in competitive environments and on means of competitive contracting. The role of reconfiguration for decommitment optimization is introduced, and an algorithm for contracting is proposed.

The motivation of our research is to explore possibilities of reconfiguring in competitive environments and to use it as a means for decommitment optimization with respect to the resource load and profit maximization.

The development of algorithms for contracting (i.e. setting of contract price and decommitment penalties) is crucial for establishing and processing a cooperation in a competitive environment. Contract setting techniques that assume both a global view of the economy and the possible business opportunities to be common knowledge among the business parties are not applicable, as such information is unavailable in a fully competitive environment. The proposed approach supports individual contract setting for each agent independently (with respect to its current state, resource load and profit), supports full agent autonomy, and corresponds to real environments.

⁶This assumption is based on the provider's knowledge of the customer's (stationary) strategy. The margins are then set in such a way the bids are no longer rejected.

The simple customer-provider scenario used in the experiments proves the presented algorithms in competitive contracting. These basic algorithms use stationary models of the customer's business strategies, and therefore they would not be able to handle adaptive behaviours of customers. Contract-setting algorithms capable of handling adaptive customer behaviours would require more complex non-stationary models of their business strategies, taking into account both long-term and short-term considerations, and also the use of approximations of the whole market situation – e.g. by building sophisticated acquaintance models of the business community. However, the proposed approach to contracting sets up a solid base for the future research on the decommitments and reconfiguration.

Acknowledgments

The research described in the paper has been supervised by Doc. Dr. Ing. Michal Pěchouček M.Sc., FEE CTU in Prague, and has been supported by the EU Integrated Projects - European Collaborative Networked Organizations Leadership (ECOLEAD - contract No. 506958) and Collaborative Process Automation Support Intelligent dynamic Agents in SME clusters (PANDA - contract No. 027169). The research has been also supported by the Ministry of Education, Youth and Sports of the Czech Republic (grant No. MSM 6840770013).

References

- [1] M.R. ANDERSSON, T.W. SANDHOLM. Leveled commitment contracts with myopic and strategic agents. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence, AAAI-98, 26-30 July 1998, Madison, WI, USA*, pages 38–45, Menlo Park, CA, USA, 1998. AAAI Press/MIT Press.
- [2] F. BERGENTI, A. POGGI, M. SOMACHER. A contract decommitment protocol for automated negotiation in time variant environments. In A. Omicini, M. Viroli, editors, *WOA 2001 - Dagli oggetti agli agenti: tendenze evolutive dei sistemi software, 4-5 September, 2001, Modena, Italy*, pages 56–61, Italy, 2001. Pitagora Editrice Bologna.
- [3] R.W. BRENNAN, M. FLETCHER, D.H. NORRIE. Reconfiguring real-time holonic manufacturing systems. In A.M. Tjoa, R.R. Wagner, editors, *12th International Workshop on Database and Expert Systems Applications, 3-7 Sept. 2001, Munich, Germany*, pages 611–615, Los Alamitos, CA, USA, 2001. IEEE Comput. Soc.
- [4] W. CAO, C.-G. BIAN, G. HARTVIGSEN. Achieving efficient cooperation in a multi-agent system: The twin-base modeling. In P. Kandzia, M. Klusch, editors, *Cooperative Information Agents*, number 1202 in LNAI, pages 210–221. Springer-Verlag, Heidelberg, 1997.
- [5] J. COLLINS, B. YOUNGDAHL, S. JAMISON, B. MOBASHER, M. GINI. A market architecture for multi-agent contracting. In *Proceedings of the Second International Conference on Autonomous Agents, 9-13 May 1998, Minneapolis, MN, USA*, pages 285–92, New York, NY, USA, 1998. ACM.
- [6] T. COUDERT, P. BERRUET, J.-L. PHILIPPE. Integration of reconfiguration in transitic systems: an agent-based approach. In *SMC '03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics, 5-8 Oct. 2003, Washington, DC, USA*, volume 4, pages 4008–4014, Piscataway, NJ, USA, 2003. IEEE.
- [7] B. DUNIN-KEPLICZ, R. VERBRUGGE. Evolution of collective commitment during teamwork. *Fundamenta Informaticae*, 56(4):563–592, August 2003.
- [8] C.B. EXCELENTE-TOLEDO, R.A. BOURNE, N.R. JENNINGS. Reasoning about commitments and penalties for coordination between autonomous agents. In *Proceedings of the Fifth International Conference on Autonomous Agents, 28 May-1 June 2001, Montreal, Que., Canada*, pages 131–8, New York, USA,, 2001. ACM.
- [9] FIPA. Foundation for intelligent physical agents [online]. (<http://www.fipa.org>), 3 2006.
- [10] J. III HAGEL, A.G. ARMSTRONG. *Net Gain-Expanding Markets through Virtual Communities*. Harvard Business School Press, Boston, MA, USA, 1997.
- [11] E. INOHIRA, A. KONNO, M. UCHIYAMA. Layered multi-agent architecture with dynamic reconfigurability. In *IEEE ICRA 2003 Conference Proceedings. 4-19 Sept. 2003, Taipei, Taiwan*, volume 3, pages 4060–4065, Piscataway, NJ, USA, 2003. IEEE.
- [12] H.J. LEVESQUE, P.R. COHEN, J.H.T. NUNES. On acting together. In *AAAI-90 Proceedings, Eighth National Conference on Artificial Intelligence, 29 July-3 August 1990, Boston, MA, USA*, volume 2, pages 94–99. Cambridge, MA, USA, 1990. MIT Press.
- [13] V. MARIK, M. PECHOUEK, J. VOKRINEK, A. RIHA. Application of agent technologies in extended enterprise production planning. In *EurAsia-ICT 2002: Information and Communication Technology*, pages 998–1007, Berlin, Germany, 2002. Springer.
- [14] V. MAŘÍK, M. PĚCHOUČEK, O. ŠTĚPÁNKOVÁ. Social knowledge in multi-agent systems. In M. Luck, V. Mařík, O. Štěpánková, editors, *Multi-Agent Systems and Applications*, LNAI. Springer-Verlag, Heidelberg, 2001.
- [15] M. PECHOUEK, V. MARIK, J. BARTA. A knowledge-based approach to coalition formation. *IEEE Intelligent Systems*, 17(3):17–25, 2002.
- [16] M. PECHOUEK, J. VOKRINEK, P. BECVAR. Explantech: Multi-agent support for manufacturing decision making. *IEEE Intelligent Systems*, 20(9):67–74, 2005.
- [17] M. PĚCHOUČEK, V. MAŘÍK, O. ŠTĚPÁNKOVÁ. Role of acquaintance models in agent-based production planning systems. In M. Klusch, L. Kerschberg, editors, *Cooperative Information Agents IV - LNAI No. 1860*, pages 179–190, Heidelberg, July 2000. Springer Verlag.
- [18] N.M. SADEH, D. HILDUM, D. KJENSTAD, A. TSENG. Mascot: An agent-based architecture for dynamic supply chain creation and coordination. *Production Planning and Control*, 12(3), 2001.
- [19] T.W. SANDHOLM, V.R. LESSER. Leveled commitment contracts and strategic breach. *Games and Economic Behavior*, 35(1-2):212–70, April-May 2001.
- [20] J.M. SWAMINATHAN, S.F. SMITH, N.M. SADEH. Modeling supply chain dynamics: A multiagent approach. *Decision Sciences*, 29(3):607–632, 1998.
- [21] WITTIG T. *ARCHON: An Architecture for Multi-agent System*. Ellis Horwood, Chichester, 1992.
- [22] P.J. 'T HOEN, J.A. LA POUTRE. A decommitment strategy in a competitive multi-agent transportation setting. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 03, Melbourne, Vic., Australia*, pages 1010–1011, 2003.

About Authors...

Jiří BÍBA is currently a PhD student in Artificial Intelligence and Biocybernetics and works as a researcher in the Agent Technology Group of Gerstner Laboratory at the Faculty of Electrical Engineering, CTU in Prague. His research focuses on issues of reconfiguration, cooperative problem solving and optimization of decommitments in competitive multi-agent environments. Currently, he is involved in research on agent technologies supporting the integration process in the European ERP/CRM industry (IST Project PANDA).

Jiří VOKŘÍNEK is currently a PhD student in Artificial Intelligence and Biocybernetics and works as a researcher in Agent Technology Group of Gerstner Laboratory at the Faculty of Electrical Engineering, CTU in Prague. His research focuses on issues of planning and replanning in manufacturing, virtual organizations and supply-chain management. Currently, he is involved in research on advanced agent-based technologies for supporting virtual organizations (IST Integrated Project ECOLEAD) and agent technologies supporting the integration process in the European ERP/CRM industry (IST Project PANDA).