Agent Contracting and Reconfiguration in Competitive Environments

Jiří Bíba, Jiří Vokřínek
Gerstner Laboratory, Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Technicka 2, 166 27 Prague 6, The Czech Republic
{biba,vokrinek}@labe.felk.cvut.cz

Abstract

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

1 Introduction

Multi-agent system (MAS) is a widely used paradigm for the modelling, planning and control of various processes. Generally, it uses distributed negotiation techniques for achieving particular goals. Besides standard centralized planning and optimization mechanisms, the MAS supports local replanning with minimal needed changes of the entire plan. There are several MAS implementations for production planning – e.g. [Pechoucek *et al.*, 2005] and for cooperation across the supply chains [Sadeh *et al.*, 2001; Swaminathan *et al.*, 1998; Marik *et al.*, 2002]. The modern business speeds up the research in the domain of Virtual Organizations [Hagel and Armstrong, 1997] that transform supply chains into dynamic cooperative networks[1]. Cooperation in such environment is based on a distributed negotiation of individual partners that leads to a satisfaction of individual or common goals. In case of an internal cooperation (within an enterprise), the goal is to maximize the overall profit of the whole enterprise. Changing the scope to an external cooperation (across a supply chain), the behaviour of the parties involved is more self-interested as their goals are maximizations of their own profits. Standard negotiation protocols and techniques used in MAS do not follow this course, so new negotiation principles for such environment have to be investigated.

2 Competitive and Collaborative Environments

Let us introduce a difference between a *collaborative* and a *competitive* multi-agent environments [Andersson and Sandholm, 1998]. 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. In contrary, by a *competitive* multi-agent environment we understand an agent community, where the primary motivation of the agents is a maximization of their individual utilities, no matter what the social welfare of the community is (agents are so called *self-interested*). The agents establish a cooperation on the process of achieving a common goal only if it contributes to the maximization of their individual utilities.

As showed before, a cooperation across a supply chain differs substantially from the one within an enterprise. We distinguish collaborative and competitive environment and inspect different aspects of cooperation in these cases. In our work we focus mainly to reconfiguration and replanning as the crucial part of the successful agents' cooperation. This paper shows

[1]The cooperation with the other partners of a Virtual Organization allows the enterprise to react on incoming business opportunities which could not be covered by the enterprise alone.

that adjusting of the contract in a real-world setting is to be analyzed. Proper algorithms for contracting have to be developed as they underlie the cooperation in a competitive environment.

3 Commitments and Decommitments

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

Whereas, for contracting in collaborative environments there is usually no need of any explicit metrics of the individual utility or the social welfare gained[2], in competitive environments an explicit expression of utility is desirable. It facilitates implementation of the rewards and penalties as the utilities that the agents gain or loose. A concept of such explicit utility evaluation is then a part of commitments – an agent providing a service (*contractee*) commits not only to perform appropriate actions (in order to gain the utility promised – it is the agent's motivation), but to provide a compensation if fails (e.g. a compensation of the profit lost to the other party). Simultaneously, the other party (*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 a profit loss and is paid e.g. the opportunity cost).

3.1 Levelled Commitment Contracts

The most complete approach to the commitments in a competitive environment has been presented by Sandholm and Lesser [Sandholm and Lesser, 2001] as *levelled commitments*. The levelled commitments include an explicit utility evaluation in a form of a *contract price* and *penalties*. They facilitate a decommitment that was not acceptable for *full commitments* commonly used. A full commitment is defined by a contract obligation as a n-tuple (Γ, ρ), where Γ introduces a description of what each of the two parties (contractor and contractee) has to perform (handling tasks, contributing goods, lending resources, etc.) and ρ introduces a contract price that the contractor has to pay to the contractee. Neither of the agents may drop the commitment under any circumstance till it is brought to a good end. On contrary, *levelled commitments* are defined as a n-tuple (Γ, ρ, a, b), where the extending parameters a and b introduce penalties to be paid in cases of decommitments.

The levelled commitments are based on a non-cooperative game theory. A negotiation process consists of two parts – (i) *contracting game*, when the agents agree on a contract and (ii) *decommitting game*, when they decide whether or not to decommit. There may occur various events (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 the knowledge of *ex ante* probability density functions (p.d.f.) of receiving the best outside offers. The p.d.f. are assumed to be a common knowledge between the contractor and the contractee.

The levelled commitments have several limiting assumptions that facilitate equilibrium calculations of the contract settings, however, make the use the of levelled commitments more difficult the 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 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 the last one (iii) [Excelente-Toledo *et al.*, 2001].

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

3.2 Contract Setting

An extension of the levelled commitment contracts have been introduced by Excelente-Toledo *et al.* in [Excelente-Toledo *et al.*, 2001]. There are provided both an algorithm for the calculation of the contract setting and an algorithm for considering a decommitment. Unfortunately, the assumptions considered (e.g. omitting the fixed costs) does not need to be always acceptable. Thus, the algorithms for the contract setting in the competitive environments are to be developed.

A price of a contract in a real world covers at least

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

the following three items: (i) *variable costs* that depend on a contract size, feasibility issues, etc. (i.e. specific conditions related to a particular contract), (ii) *fixed costs* that are not related to a particular contract, but are related to the overall business and are 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 of the enterprise owner). A penalty in a real world seeks to cover at least a portion of fixed costs and also the *profit lost*. While the calculation of variable and fixed costs, resp. profit lost, is rather pragmatic, the setting of the intended profit is rather strategic or even speculative with the dependence on many aspects (e.g. experience with the second party, various social relations, the first-party profit eagerness 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. 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 w.r.t. contract size is 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 costs[3]. The service provider does not try to get more than to cover all the costs (variable and fixed). It calculates variable costs and projects its actual fixed costs to be covered to the margin[4]. 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 w.r.t. the variable costs – let say 50% of the variable costs

learned safe limitation – the margin is limited based on the past experience to an average value of the previously accepted margins; if no previous margin was accepted, the new margin is set to a half of the minimal rejected margin from the past

learned speculative margin – the margin is limited based on the past experience to mid between the maximal accepted and minimal rejected margin from the past; if no previous margin was accepted, the margin is set 20% less than the minimal rejected margin from the past; if no previous margin was rejected, the margin is set 50% bigger than the maximal accepted margin from the past (the asymmetric decrease/increase rates were set empirically in order to gain a quick algorithm convergence); the speculation stops if the increase of the margin w.r.t the previous maximal accepted margin is less than a certain value – let say 1% and the margin is then always set to the maximal previous accepted margin

Obviously, the first approach does not guarantee the provider a profit in a long-term view. The other two approaches do, because the provider learns from the past. The latter limitation then promises the maximal possible profit (experiments are provided in the section 4).

3.3 Reconfiguration in Competitive Environments

Even though there are some domain-dependent application implementations of a reconfiguration in multi-agent systems (MAS) - e.g. [Coudert *et al.*, 2003; Inohira *et al.*, 2003; Brennan *et al.*, 2001], the first deep study of a reconfiguration and its formalization was published merely recently by Dunin-Keplics in [Dunin-Keplicz and Verbrugge, 2003]. The multi-agent environment has been assumed to be collaborative. Thus, if a failure occurred, then all the agents involved in achieving their common goal made their best for establishing a recovery in order to complete their task successfully (a reconfiguration occurred). This behaviour has been given by their persistent collective intention set to achieve their common goal in accordance to a definition of a joint persistent goal [Levesque *et al.*, 1990].

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). A decommitment as a concept was not considered in the above-mentioned research, while according to ['T Hoen and La Poutre, 2003] it may become a means for the optimization of agents' individual profits.

While an 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, the decommitment or replanning is driven by the common goal. It is obvious that both partners of the 'contract' have the same motivation to keep it or change it. In 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 it is necessary. A self-interested agent will be rather reluctant to take on further obligations if it were not rewarded or if it even entailed its profit loss.

One of the motivations at hand is to use the reconfiguration as an alternative to a decommitment if keeping the actual contract does not contribute to the maximization of the agent's individual utility (e.g. a more profitable offer appeared). For example, the agent may find a subcontraction being reimbursable from the reward promised by its contractor and decides to take advantage of both, the actual and the new contracts without decommitting. Of course, the

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

[4] The fixed costs are constant for a time unit, but generally are accumulated or reduced based on whether the provider's business was successful in the past or not.

considering the decommitment might have started also on account of unexpected events, e.g. resources failure, unexpected delay that might put at risk the meeting of the contract deadline, etc. Thus, a reconfiguration may appear to be useful for the optimization of the resources load and also profit maximization by means of optimizing eventual or necessary decommitments. In collaborative environment the reconfiguration is usually driven by individual tasks (it is invoked top-down by decomposition means) rather than by intention of the contracted agents. Table 1 shows an overview of the different environment properties in several aspects.

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

Table 1: Overview of the different environment properties in several aspects

Let us provide an example of a reconfiguration and its potential. A customer *Customer 1* grants a contract *Contract 1* to a group of agents – coalition [Pechoucek *et al.*, 2002] – coordinated by a provider *Provider Leader*. While the contract is being executed, a coalition member *Provider Traitor* receives a proposal on 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*. Still, it may save the decommitment penalty if it finds a subcontractee *Provider Subcontractee* which takes on its obligations.

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

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

In any case, the reconfiguration strongly depends on the contract setting and vice versa. This is 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 firing by guess than a deliberative calculation. The implementation of a more sophisticated contracting algorithm capable of a reconfiguration and also handling e.g. customer's adaptive preferences about the margin acceptance boundary are currently under research.

4 Experiments

The experiments provided show properties of our algorithm (in three variants) as it was proposed in section 3.2. Our model implementation defined two types of agents – the *customer* and *provider*. Sizes of contracts proposed by the customers were arbitrary within a bounded interval, their number varied from none to several per day and 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 was performed 50 runs of the simulation and the results were averaged[5]. Let us describe the results in detail. The Fig. 1 introduces a comparison of the margin setting for all the three algorithm variants with respect to the customer's margin boundary.

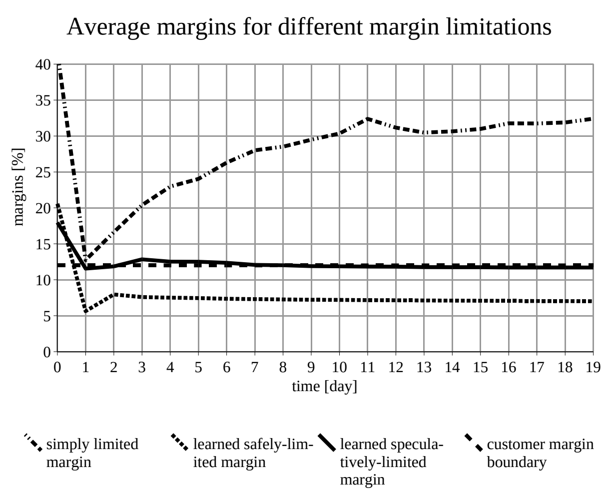

Figure 1: Comparison of all the proposed limitations: average margins and customer's margin boundary

The first variant of the algorithm implements the simple limitation of margins. As the agent does not take into account the past experience and its only concern is a coverage of its actual fixed costs, the success

[5]Due to both the randomness of the contract sizes and the absence of the 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.

ratio (a ratio of the number of successful contracts to the number all contracts) decreases together with the growth of the debts reflected in the margin set (see figure 2). Although sometimes arrives a contract 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), it happens seldom only 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 in debts, there is only a little probability that it gets a chance to pay them back.

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

The third variant implements the learned speculative limitation of margins. In the beginning of the simulation the provider speculates and tries to learn each customers' margin boundary. Thus, it is less successful in the beginning. On the other hand once it approaches as good setting of the margins as possible (and reasonable), it becomes to be better off on a benefit from a turnover. The learned margin guarantees acceptance of all the bids and also maximal pay-off on the particular customer. Running into debts may also occur in this case if the customers' margins are too low and neither the maximal benefit from the turnover covers the provider's fixed costs. On the other hand, the provider cannot defend himself against this situation and it is obvious, that its running in debts is as slow as possible.

The Fig. 2 and 3 introduce a comparison among all the three algorithm variants. While in the first variant the provider runs in debts in the second and the third one becomes to be better off. Although the average success ratio of the third variant grows slower in the beginning than the success ratio of the second one (due to the speculation in the beginning) the account balance shows that the gained profit was worth entailing the loss in the beginning. Let it be noted that the success ratio was computed w.r.t. all the simulation time, i.e. the third-variant success ratio would

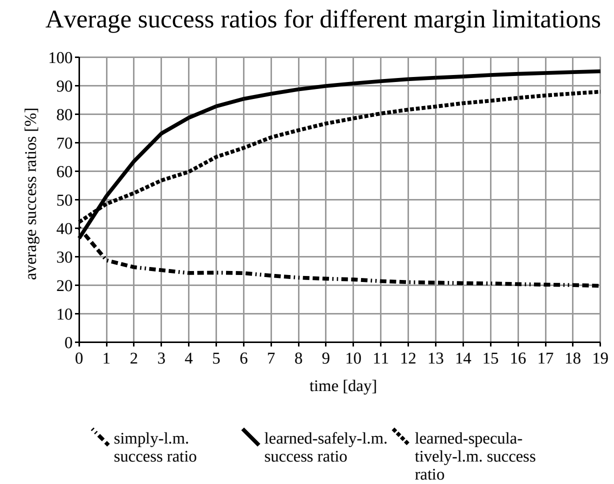

Figure 2: Comparison of all the proposed limitations: average success ratios

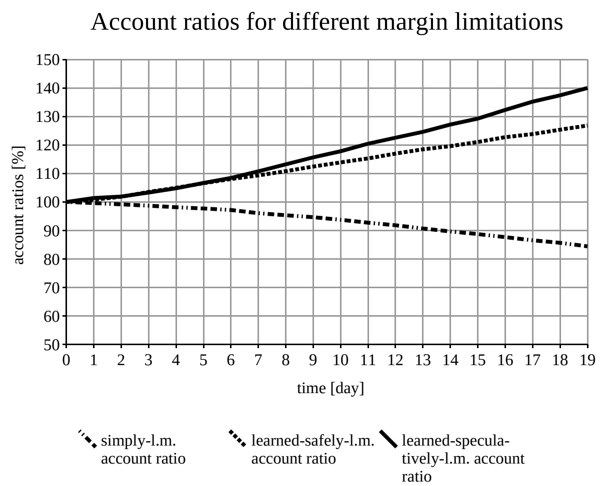

Figure 3: Comparison of all the proposed limitations: account ratios

converge to the second-variant one that converges to 100% in an infinite time horizon[6].

5 Conclusion

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

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

The development of algorithms for contracting (i.e. setting of a contract price and decommitment penal-

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

ties) is crucial for establishing and processing of a cooperation in a competitive environment. Contract setting techniques assuming both the global view of the economy and the possible business opportunities to be a 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 a full agent autonomy and corresponds to real environments.

Simple customer-provider scenario used in experiments proves presented algorithms in a competitive contracting. These basic algorithms use stationary models of the customer business strategies, and therefore they would not be able to handle adaptive behaviours of the 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 the long-term and short-term aspects and also using approximations of the whole market situation – e.g. by building acquaintance models of the business community. However, the proposed approach to contracting sets up the solid base for the future research aimed to the decommitments and reconfiguration.

Acknowledgments

This research work has been supported by the EU Integrated Project, European Collaborative Networked Organizations Leadership (ECOLEAD) and the Ministry of Education, Youth and Sports of the Czech Republic grant No. MSM 6840770013.

References

[Andersson and Sandholm, 1998] M.R. Andersson and 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.

[Brennan *et al.*, 2001] R.W. Brennan, M. Fletcher, and D.H. Norrie. Reconfiguring real-time holonic manufacturing systems. In A.M. Tjoa and 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.

[Coudert *et al.*, 2003] T. Coudert, P. Berruet, and 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.

[Dunin-Keplicz and Verbrugge, 2003] B. Dunin-Keplicz and R. Verbrugge. Evolution of collective commitment during teamwork. *Fundamenta Informaticae*, 56(4):563–592, August 2003.

[Excelente-Toledo *et al.*, 2001] C.B. Excelente-Toledo, R.A. Bourne, and 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.

[Hagel and Armstrong, 1997] J. III Hagel and A.G. Armstrong. *Net Gain-Expanding Markets through Virtual Communities*. Harvard Business School Press, Boston, MA, USA, 1997.

[Inohira *et al.*, 2003] E. Inohira, A. Konno, and 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.

[Levesque *et al.*, 1990] H.J. Levesque, P.R. Cohen, and 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.

[Marik *et al.*, 2002] V. Marik, M. Pechoucek, J. Vokrinek, and 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.

[Pechoucek *et al.*, 2002] M. Pechoucek, V. Marik, and J. Barta. A knowledge-based approach to coalition formation. *IEEE Intelligent Systems*, 17(3):17–25, 2002.

[Pechoucek *et al.*, 2005] M. Pechoucek, J. Vokrinek, and P. Becvar. Explantech: Multiagent support for manufacturing decision making. *IEEE Intelligent Systems*, 20(9):67–74, 2005.

[Sadeh *et al.*, 2001] N.M. Sadeh, D. Hildum, D. Kjenstad, and A. Tseng. Mascot: An agent-based architecture for dynamic supply chain creation and coordination. *Production Planning and Control*, 12(3), 2001.

[Sandholm and Lesser, 2001] T.W. Sandholm and V.R. Lesser. Leveled commitment contracts and strategic breach. *Games and Economic Behavior*, 35(1-2):212–70, April-May 2001.

[Swaminathan *et al.*, 1998] J.M. Swaminathan, S.F. Smith, and N.M. Sadeh. Modeling supply chain dynamics: A multiagent approach. *Decision Sciences*, 29(3):607–632, 1998.

['T Hoen and La Poutre, 2003] P.J. 'T Hoen and 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.