KrokodilAgent: A Supply Chain Management Agent

A. Petrić, K. Jurasović
Faculty of Electrical Engineering and Computing / Department of Telecommunications, Zagreb, Croatia
{ana.petric, kresimir.jurasovic}@fer.hr

***Abstract*— The Trading Agent Competition (TAC) promotes research in the trading agent problem. TAC has two competitive scenarios. First and the older one is TAC Classic where 8 agents compete by assembling travel packages for customers with different preferences for the trip. The second one is Supply Chain Management (SCM) where 6 agents compete by assembling computers and selling them to customers. We decided to join the TAC SCM game because we found it to be more challenging and complex then the TAC Classic. In this paper we present KrokodilAgent, our entry in the TAC SCM 2004.**

I. INTRODUCTION

Supply chain management involves several activities like raw material procurement, producing, selling and delivering finished goods. Supply chains have an important role in today's global economy. The purpose of SCM game is to explore how to maximize profit given the conditions that dominate on the market. It is also important to establish how the changes during the game effect on the game outcome.

In the TAC SCM game scenario each of the six agents has its own PC manufacturing company. During the 220 TAC days agents compete in two different markets. On the first market agents compete by buying raw materials necessary to produce personal computers.

Participants on the first market are agents and eight suppliers that produce four types of components (CPUs, motherboards, memories, hard drives) with different performances. In his factory the agent can manufacture 16 types of PCs.

On the second market the agents are trying to sell all the PCs they produced to customers and at the same time earn as much money as possible. The winner is the agent with the highest bank account at the end of the game.

II. GAME OVERVIEW

To play in the game an agent has to connect to the game server. The server has multiple functionalities, it simulates customers and suppliers, controls agent's factory and warehouse and runs the bank. The server is shown in Figure 1.

Each agent has an account in the bank and every day he gets the report with his current bank balance. At the beginning of the game agent has no money so he has to loan money from the bank. For every day that the agent is in depth the bank charges him interest and for every day that his bank account is positive the bank pays interest to the agent.

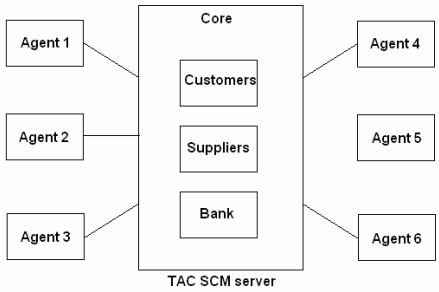

Figure 1. TAC SCM architecture

There is no upper limit how much money can an agent loan from the bank. That is not good because a destructive agent can buy all of the components with no intent of using them and make it impossible for other agents to produce PCs. It is also very unrealistic because in the real world no bank is going to grant anyone such big loans without any cover.

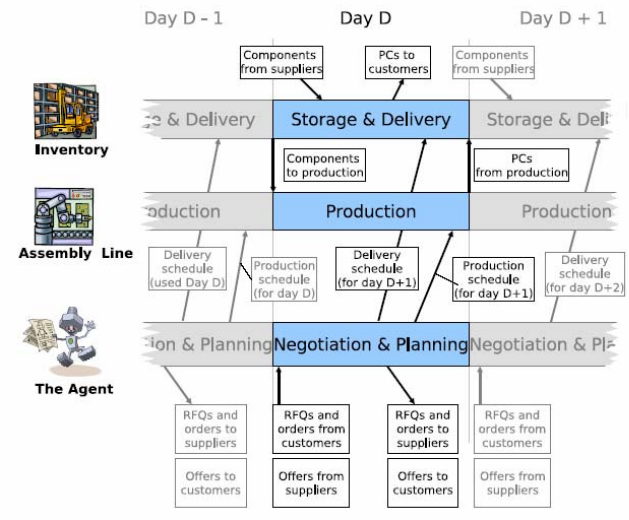

Figure 2. Illustration of a TAC SCM day [1]

Agent's daily responsibilities are divided into four logical tasks that are described in the next sections.

A. Negotiate supply contracts

In order to sell PCs it is necessary to purchase components and produce PCs from those components. A very popular approach is to purchase a large amount of components at the beginning of the game. This strategy was first used in TAC SCM 2003 competition and became very popular but soon it started to cause problems known as the day-0 effect.

It became popular thanks to the formula that calculates component prices. At the beginning of the game there is no demand for components and all of the supplier capacity is free, that results with low component prices. During the game demand for components rises and supplier usually has little or no free capacity so component prices constantly rise.

At the beginning of each day agent gets the offers for components as the result of Request for Quotes (RFQs) that he sent to suppliers the day before. Suppliers also deliver the components that the agent ordered earlier, those components can't be used for production on that day. Agent can send maximum of ten RFQs to each supplier every day. The upper limit is ten so that suppliers wouldn't be swamped with RFQs, especially at the beginning of the game.

Supplier bundles all agent RFQs in an *ordered* list descending by likelihood. Agent's *likelihood*, *weight* and *order ratio* for every supplier are calculated as:

$$\text{Order ratio} = \frac{\text{Quantity Purchased}}{\text{Quantity Requested}},$$

$$\text{Weight} = \min(0.5, \text{Order ratio}),$$

$$\text{Likelihood}_{\text{A}} = \frac{\text{Weight}_{\text{A}}}{\sum_{agents} \text{Weight}}.$$

Quantity Requested is the sum of quantities in all RFQs that an agent sent to the supplier from the beginning of the game. *Quantity Purchased* is the total quantity that an agent purchased from the supplier from the beginning of the game. At the beginning of the game *order ratio* of all agents is 1 so that all agents have the same likelihood on the day 0. Because of that supplier randomly sorts all agent RFQs that he receives on the day 0 and sends the offers in that random order, so if the agent is lucky that his RFQ was the first in line he will get his components by the day he requested them, otherwise he probably won't be able to get any components until later in the game.

Another problem is that the agent has to send a new RFQ for every of ten different components. Since all four types of components are necessary to produce a PC and the probability of getting all of the components first is very small, in the first part of the game agents just wait for their components to be delivered.

During the game the agent that sends RFQs with large amounts and then doesn't buy requested components later in the game has trouble with component purchase because his reputation is low. Sorting RFQs and sending offers by agent's likelihood is there to prevent agents who are trying to raise component prices for other agents or block other agents from purchasing components.

If the supplier can't deliver the requested quantity of components by the requested due date, he sends two types of offers to the agent. First offer is called a partial offer and it contains only a part of the requested quantity that can be delivered on the due date. The second offer is called an earliest complete offer, in it the supplier offers the requested quantity to be delivered the earliest day as possible in regard to his production capabilities.

In TAC SCM 2003 and TAC SCM 2004 agent sends RFQs to suppliers that contain type of the component, requested quantity of the component and the due date when he wants the components to be delivered. In TAC SCM 2005 besides component type, quantity and due date agent has to specify the reserve price he is willing to pay for the components. This is one of the biggest changes in the game rules, it should eliminate day-0 effect and partially change the way that agents purchase components.

Other improvements on the supplier side include new ways of determining available supplier capacity, calculating offer prices for components, allocating factory capacity and calculating agent's reputation.

Another way of eliminating the day-0 effect would be to determine a limit on how high the agent's negative bank balance can be. If the permitted negative balance is low the agent won't be able to buy large amount of components.

B. Bid for customer orders

At the beginning of each day agent gets customer RFQs that contain type of the requested PC, quantity of PCs, due date for the PC delivery, penalty for late delivery and the reserve price that the customer is willing to pay for every PC. After analyzing all of the customer RFQs agent sends offers for those RFQs he considers to be profitable and that can be fulfilled by the specified due date.

The offer specifies price of the PC, quantity and due date. The customer will consider the offer if it contains entire quantity specified in the RFQ, if the delivery of the PCs is on due date specified in the RFQ and if the offer price is below or equal to the reserve price specified in the RFQ. From all considered offers customer selects the offer with the lowest offer price and sends the order for the PCs to the agent that sent the most favorable offer.

Agent has to deliver the PCs day before the due date because it takes one day for the PCs to arrive to the customers. If the agent can't deliver the PCs on time customer will charge him penalty for every day of the late delivery, after the fifth day customer cancels the order and stops charging penalties if the PCs haven't arrived in the meantime.

C. Manage daily assembly activities

Agent assembly cell capacity during the game is 2000 cycles per a day and unlike the supplier capacities it doesn't fluctuate. Every day agent receives an inventory report form his factory. The report contains quantities of

components available for production and finished PCs available for delivery.

After receiving the report agent decides how to allocate those available components and free factory capacity in order to produce PCs, he sends this decision to the factory in form of the daily production schedule. The production schedule is always sent for the next day. The PCs listed it the schedule are produced the next day and shipped into the warehouse, they can be sent to the customers the day after.

For keeping components and PCs in the warehouse every day is charged a storage cost which is a percentage of the component base price. In last two years storage cost had a large growth. The game authors considered that drastically increasing storage cost will lower down the day-0 effect because keeping the components in the warehouse raises the component expenses. After some time it became obvious that the impact of raising storage cost is not that significant so the agents continued to use the day-0 strategy.

PCs are assembled from four types of components: CPUs, motherboards, memories and hard drives. CPUs are produced by two suppliers called Pintel and IMD, they come in two speeds: 2.0 and 5.0 GHz. Pintel CPUs work only with Pintel motherboards and IMD CPUs work only with IMD motherboards. There are two producers of motherboards called Basus and Macrostar, they both produce Pintel and IMD motherboards. Memory is produced by MEC and Queenmax in sizes: 1 and 2 GB. Hard discs come in two sizes: 300 and 500 GB, they are produced by Watergate and Mintor.

These ten different components can be combined into 16 different PC configurations. PCs are divided into three market segments: Low range, Mid range and High range. PCs in the High range have the best components and they are worth more but their production requires more assembly cycles.

D. Ship completed orders to customers

If the agent sends the production schedule to the factory on the day *d*, the PCs listed in it are produced on the day *d+1* and on the same day they are moved into the inventory. If the agent sends the delivery schedule on the day *d+1* the PCs can be sent to the customers on the day *d+2*, and the customer will receive them on the day *d+3*. The agent fulfills customer's order by delivering requested PCs. Customers pay received PCs on the due date or the day after receiving the PCs, whichever is later.

III. KROKODILAGENT

KrokodilAgent is an intelligent agent developed at Department of Telecommunications, Faculty of Electrical Engineering and Computing in Zagreb, Croatia. Agent is divided into five logical units: CustomerImpl, FactoryImpl, InventoryImpl, SupplierImpl and ZTELAgent.

CustomerImpl takes care of the customer orders, calculates profit and the offer prices for PCs. FactoryImpl organizes factory utilization during the game. InventoryImpl orders new components when necessary, checks is there enough components for a specific PC offer, calculates component and basic PC prices. SupplierImpl responds to supplier offers and takes care of component delivery. ZTELAgent takes care of the begging and the end of the game, simulation status and responds to customer RFQs.

KrokodilAgent's basic functions are described in the next chapters.

A. Negotiate supply contracts

On the day 0 the agent orders components worth 50mil $. Ordered components are delivered several times during the game. In case that the agent didn't receive a satisfying offer he sends new RFQs on the first day for those components and delivery dates he didn't get on the day 0.

In case that manufacturing and selling PCs is very successful so that all the components were spent and the ones ordered on day 0 are not going to arrive soon, the agent orders additional components. In the games with successful competitors additional ordering is very rare.

Experimentally we determined two limits that are considered when the agent orders components during the game. The upper limit is 800 units for CPUs and 1600 for other components, and the lower limit is 500 units for CPUs and 1000 for other components.

Every day agent checks the amount of components in the warehouse. He establishes the amount of components that are available for production the next day and decides is it necessary to order new components for further production.

If the agent has more components then the upper limit he won't order anything that day. If the number of components is between those two limits the agent orders the amount of components he spent on production that day, and if the number of components is under the lower limit the agent orders the amount of components he spent on production that day enlarged by the number of components necessary to reach the lower limit.

B. Bid for customer orders

When the agent receives RFQs he calculates the average price for every component type and basic PC price. The agent always knows how much did he pay for every component that is currently in the warehouse so he can calculate the average component price. Basic PC price is calculated from the average prices of every component incorporated in the PC. To get the offer price for the PC the agent adds his margin on the basic PC price.

The margin varies during the game. If the demand for PCs is high and the agent has already won a large amount of orders it means that he is offering PCs at low prices. In that case he increases the margin because he is spending components for low profitable PCs and is not making enough money. If the demand is low and the agent isn't getting any orders he lowers the margin in order to make some money and pay off the components that he already had to pay when they were delivered.

Agent's factory utilization also has an impact on the margin. If the utilization is higher then the 75% the margin is enlarged, and if it is lower then the 45% the

margin is decreased. At the end of the game the margin is 0 because the only goal at the end is to sell out the rest of the components in the warehouse.

After analyzing customer RFQs the agent sends offers to customers for all RFQs that he can produce from the components he currently has in the warehouse. He also carefully considers the RFQs with earlier delivery dates so that he would have enough time to assembly the PCs.

C. *Manage daily assembly activities*

Since the PCs are produced after the agent receives the customers order it is very important to organize efficient production. The agent keeps track of free factory capacity, if the factory has no free capacity in the next *d* days he doesn't send offers for those RFQs which have to be delivered in the next *d* days. This condition is important because if the agent gets to many orders he probably won't be able to produce all the necessary PCs on time and that will result with high penalties. After the agent receives the list of orders won that day he creates the production schedule so that the PCs with earlier due dates are produced first.

D. *Ship completed orders to customers*

PCs are shipped to the customer right after they are produced so that the agent wouldn't have to pay storage cost for them.

IV. TAC SCM 2004

TAC SCM games in the year 2004 were held on two servers. The competition was divided into three parts: Qualifications held from 7.-18. June, Seeding held from 5.-16. July and the Finals held from 20.-22. July.

Qualifying and Seeding Rounds were played on servers `tac3.sics.se` and `tac4.sics.se.` There were 31 teams competing in the Qualifying and 29 teams in Seeding Rounds. In the Finals 24 teams were competing on servers `tac3.sics.se`, `tac4.sics.se`, `tac5.sics.se` and `tac6.sics.se.`

A. *Qualifying Rounds*

At the end of first week our average score was 3.163 mil $ after 21 games played on server `tac3.sics.se` and 1.274 mil $ after 21 games played on server `tac4.sics.se.`

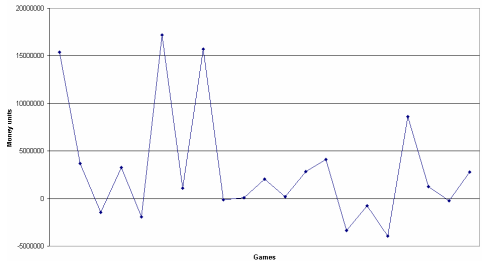

Figure 3. Qualifying Rounds - Week 1, server tac3.sics.se

In the first week we were not ordering components on day 0 because we believed that other agents won't be doing that since everybody condemned day-0 strategy on the official TAC SCM forum. After realizing that other agents are ordering large amounts of components on day 0, in the second week we also started to use that strategy combined with ordering components during the game like we described in the section *III.A*.

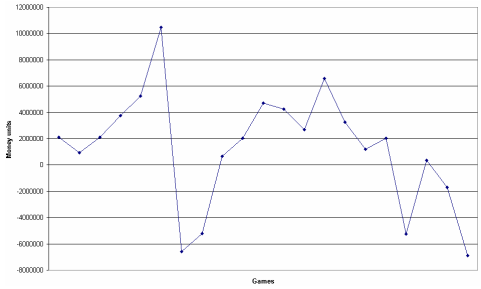

Figure 4. Qualifying Rounds - Week 1, server tac4.sics.se

Our result significantly improved in the second week, our average score was 20.769 mil $ after 18 games played on server `tac3.sics.se` and 20.165 mil $ after 18 games played on server `tac4.sics.se.`

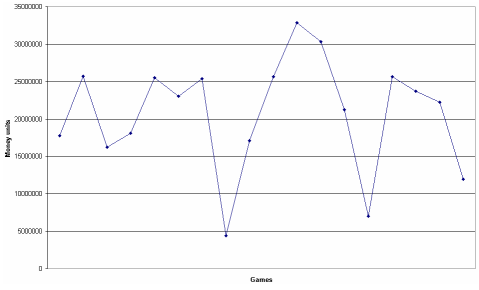

Figure 5. Qualifying Rounds - Week 2, server tac3.sics.se

At the end of the Qualifying Rounds our average scores were 11.289 mil $ on server `tac3.sics.se` and 9.993 mil $ `tac4.sics.se.` In overall our final score was 10.64 mil $ after 78 games played and that was enough to place us at 13-th position.

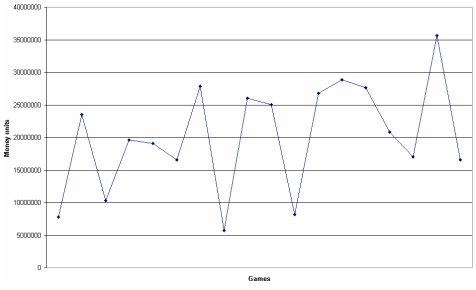

Figure 6. Qualifying Rounds - Week 2, server tac4.sics.se

B. *Seeding Rounds*

The purpose of Qualifying Rounds was to se is your agent functioning the way you want in a competitive surrounding and establish how successful it is comparing to other agents. Between Qualifying and Seeding Rounds there was a two week time period to improve your agent's

functionality based on the experience from the Qualifying Rounds.

At the end of first week our average score was –0.045 mil $ after 20 games played on server `tac3.sics.se` and 15.389 mil $ after 20 games played on server `tac4.sics.se.` The reason for the negative score is the renovation of our faculty building. We had several power and network failure that made it impossible for us to compete in 10 games and caused connection problems in one game.

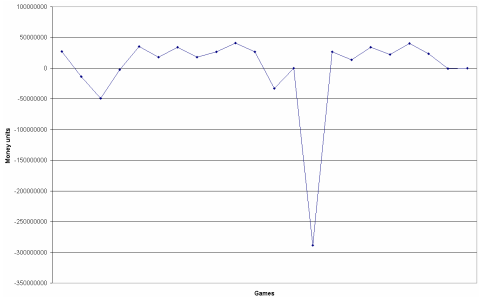

Figure 7. Seeding Rounds - Week 1, server tac3.sics.se

If the agent doesn't join the game according to the rules his result is 0 or the worst score in the game, which ever is less. When the agent connects to the game he can't recognize which day it is so he thinks it is day 0, he also can't continue to play in the game. In the game with connection problems the agent connected in the game three times and every time he connected he ordered large amount of components.

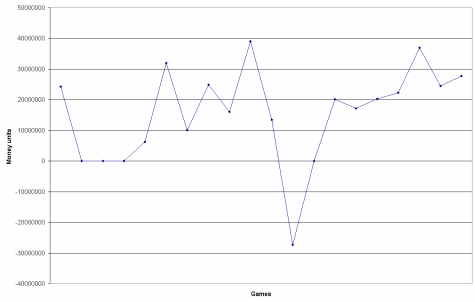

Figure 8. Seeding Rounds - Week 1, server tac4.sics.se

The average score of the games we actually played was 26 mil $ in 14 games played on server `tac3.sics.se` and 22.336 mil $ in 15 games played on server `tac4.sics.se.`

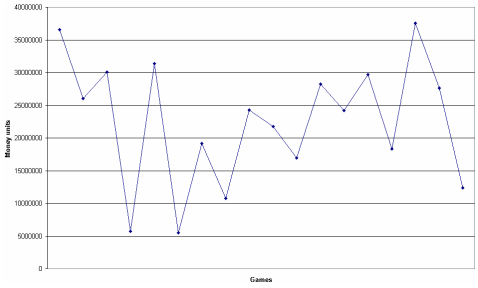

Figure 9. Seeding Rounds - Week 2, server tac3.sics.se

The second week we played without any problems similar to the ones from the first week. Our average score was 22.562 mil $ after 18 games played on server `tac3.sics.se` and 18.285 mil $ after 18 games played on server `tac4.sics.se.`

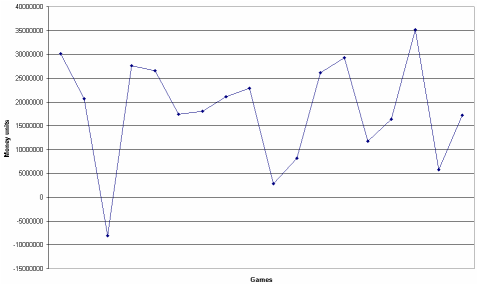

Figure 10. Seeding Rounds - Week 2, server tac4.sics.se

At the end of the Seeding Rounds our average scores were 10.663 mil $ on server `tac3.sics.se` and 16.761 mil $ `tac4.sics.se.` In overall our final score was 13.712 mil $ after 76 games played and that was enough to place us at 12-th position and qualify for the Quarter-Finals.

C. Quarter-Finals

Quarter-Finals were played in four groups: A, B, C and D. First three agents from each group qualified in the Semi-Finals. We played in group D and our opponents were: ScrAgent, GeminiJK, Socrates, Intuition and SCMAgent@CSE. There were 8 games played on the server `tac6.sics.se`, our average score was 14.66 mil $ which placed us at 2 place in our group and we qualified in the Semi-Finals.

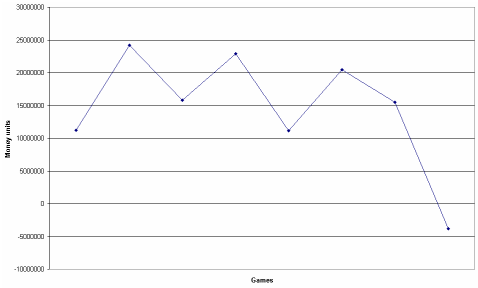

Figure 11. Quarter-Finals Group D, server tac6.sics.se

D. Semi-Finals

In the Semi-Finals ScrAgent, Socrates and KrokodilAgent played with the three best agents from the A group: FreeAgent, SouthamptonSCM and Mr.UMBC. There were 16 games played.

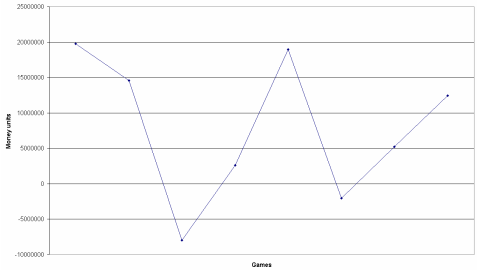

Figure 12. Semi-Finals Group 1, server tac3.sics.se

Our average score was 7.957 mil $ after8 games played on server `tac3.sics.se` and 1.707 mil $ after 8 games played on server `tac4.sics.se`. Our final score was 4.832 mil $ which placed us at the 4 place so we didn't qualify for the Finals.

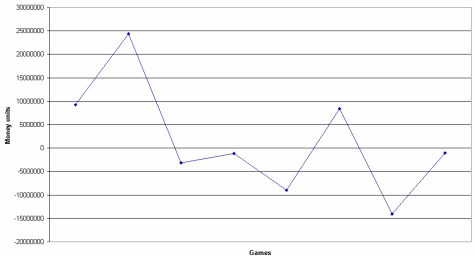

Figure 13. Semi-Finals Group 1, server tac4.sics.se

ACKNOWLEDGMENT

We would like to thank our mentors professor Ignac Lovrek and assistant professor Gordan Ježić, and the assistants Krešimir Mlinarić and Tomislav Marenić for including us into this project and supporting our work.

REFERENCES

[1] John Collinsa, Raghu Arunachalam, Norman Sadehb, Joakim Eriksson, Niclas Finne, Sverker Jansonc, "The Supply Chain Management Game for the 2004 Trading Agent Competition", *Official specification, April 2004*

[2] N. M. Sadeh, R. Arunchalam, "The 2003 Supply Chain Management Trading Agent Competition", Workshop on «Trading Agent Design And Analysis», New York, USA, 2004.

[3] N. M. Sadeh, R. Arunchalam, J. Eriksson, N. Finne, S. Janson, «TAC 03: a Supply Chain Trading Competition», *AI Magazine*, 2003.

[4] R. Arunachalam, N. Sadeh, J. Eriksson, J. Finne, S. Janson, *«Design of the Supply Chain Trading Competition»*, IJCAI-03 Workshop on «Trading Agent Design And Analysis», Mexico, 2003.

[5] K. Jurasović, A. Petrić, M. Ljubičić "Agent competition in supply chain management", Faculty of Electrical Engineering and Computing, Zagreb 2004.

[6] A. Petrić "Applying software agents in e-commerce", Faculty of Electrical Engineering and Computing, Zagreb 2005.

[7] M. Benish, A. Greenwald, I. Grypari, R. Lederman, V. Naroditski, M. Tschantz "Botticelli: A Supply Chain Management Agent", AAMAS New York, USA, 2004.

[8] P. W. Keller, F.-O. Duguay, D. Precup "RedAgent-2003: An autonomus market-based supply-chain management agent", AAMAS New York, USA, 2004.

[9] R. Dong, T. Tai, W. Yeung, D. C. Parkes "HarTAC – The Harvard TAC SCM '03 Agent", Workshop on Trading Agent Design and Analysis (TADA 04), New York, USA, 2004.

[10] J. Estelle, Y. Vorobeychik, M. P. Wellman, S. Singh "Strategic Procurement in TAC SCM: An Empirical Game-Theoretic Analysis", Workshop on Trading Agent Design and Analysis (TADA 04), New York, USA, 2004.