

CMieux Supply Chain Trading Analysis and Instrumentation Toolkit

Michael Benisch James Andrews
David Bangerter Timothy Kirchner
Benjamin Tsai Norman Sadeh

September 2005
CMU-ISRI-05-127

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This report describes the Analysis and Instrumentation Toolkit (AIT) developed by researchers at the CMU e-Supply Chain Management Laboratory for the 2005 Trading Agent Competition in Supply Chain Management (TAC SCM). It accompanies the toolkit's release to the TAC SCM research community. The tools help users to perform in-depth game analysis from any valid TAC SCM server conforming to the 2005 specification. The AIT allows users to analyze component prices, component lead times, supplier reputations, market shares, and sales of each agent that participates in the simulated supply chain management game. We provide an overview of the tools and a demonstration analysis of a game from the 2005 Finals at IJCAI.

The research that lead to the development of the software described in this document has been funded by the National Science Foundation under ITR Grant 0205435 and under IGERT grant 9972762. Several of the analysis tools were developed using an open source Java charting library, JFreeChart. We acknowledge the use of the SICS TAC SCM Game Data Toolkit in our distribution, which is Copyright (c) 2003-2005, Swedish Institute of Computer Science, all rights reserved. A binary version of this library is provided in our distribution, and the full source code and the conditions for use are available to you at <http://www.sics.se/tac/>.

Keywords: TAC SCM, Supply Chain Management, Automated Trading, Trading Visualization

1 Preface

1.1 Introduction

The Trading Agent Competition in Supply Chain Management (TAC SCM) is an international competition that revolves around a game featuring six competing agents each entered by a different team. Agents are responsible for managing the assembly of Personal Computers (PCs) while trading in dynamic supplier and customer markets. The game features a number of different types of PCs, each requiring different sets of components that can be procured from multiple suppliers. Every day customers submit Requests For Quotes (RFQs) for different types of PCs. An RFQ includes a PC type, a quantity, a delivery date, a reserve price and a tardiness penalty incurred for missing the requested delivery date. Tardiness penalties increase with the number of days by which delivery commitments are missed. Orders are simply canceled if they are excessively late. Agents also compete with one another to procure the components they need at the best possible prices, with suppliers adjusting their asking prices based on changing market conditions. Each game simulates operation of the supply chain over a period of one year, with each day being simulated in 15 seconds. Each agent has a finite capacity that limits the number of PCs it can assemble in any given day. It also has a bank account from which it needs to draw money to purchase components and pay penalties. Inventory costs are also charged against the account. Daily interests are charged on negative balances and accrued on positive ones. Agents make money by selling and delivering finished PCs to customers. Supplier and Customer market conditions are randomly change over time and from one game to another to ensure that agents are tested across a broad range of representative situations. The reader is referred to [2] for the full game specification.

One of the most important aspects of developing an entry for the Trading Agent Competition is the ability to analyze and learn from the strategies of other agents [3, 1]. Throughout the rounds of the competition each of the agents undergoes significant development in response to the strategies played by others. It was with this in mind that our team, CMieux from Carnegie Mellon University, developed an instrumentation package, AIT, to help analyze the increasingly complex strategies employed by competitors in the game.

This document accompanies the release of the AIT and provides information about how to use the various components of the toolkit. The package contains two sets of tools which can be useful for analyzing different aspects of historical game logs. The first set involves instruments that display information about Business to Business (B2B) interactions, and the second about Business to Customer (B2C) interactions. In TAC SCM the B2B interactions take place between agents and suppliers, and the B2C interactions take place between agents and customers. The B2B tools include information about agent reputations, component purchase prices, and requested lead times. The B2C tools include information about successful and unsuccessful customer bid prices, as well as detailed market share data for various game markets. We hope that making available this document and the tools themselves will assist the general TAC SCM research community.

2 Analysis and Instrumentation Toolkit Overview

Using the CMieux AIT allows users to accomplish analysis tasks such as,

- **Gathering Data:** the AIT will automatically download, and parse game files from any 2005 (or earlier) TAC SCM Server. These files are cached locally on disk for easy re-analysis of the same games. Users need only specify the game number, and server, the rest of the process is fully automated.
- **Focusing Analysis:** users can choose to focus analysis on a particular agent in the game by highlighting its performance in several of the included tools. Focusing allows users to easily highlight the performance of a particular agent rather than sifting through excess visual data.
- **Graphing Supplier Reputations:** the Supplier Reputation tool (described in detail in Section 5.1, shown in Figure 1) graphs agent reputation for all suppliers over the entire game. Supplier reputations change over the course of a game depending on agent interactions with suppliers, and viewing this data graphically allows users to easily pick out interesting areas.

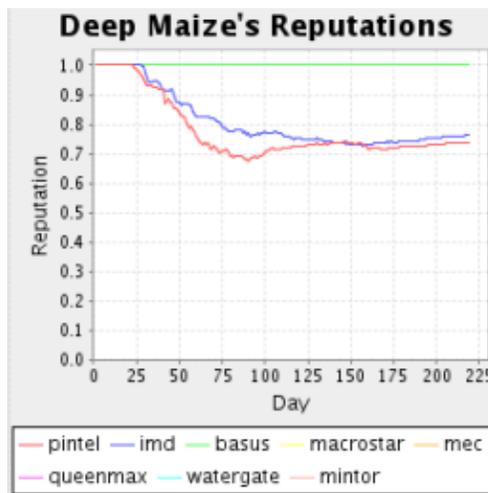


Figure 1: Supplier Reputation Graph

- **Graphing Agent Lead Times:** users can display agent lead times, and their associated prices and quantities graphically for all supplier orders using the Supplier Lead Time tool (described in Section 5.3, shown in Figure 2). Supplier lead times play a large role in determining an agent's material costs, our tool allows users to view the effects of the various lead time strategies used simultaneously in a graphical manner.
- **Graphing Component Prices:** the Component Sales tool allows users to graph weighted average daily component prices of all agents over the entire game by component type (described in Section 5.2, shown in Figure 3). This can help users identify what price a particular agent was paying for supplies relative to the other agents at different points during the game. Once a period of the game is identified as being problematic users can switch to the Supplier

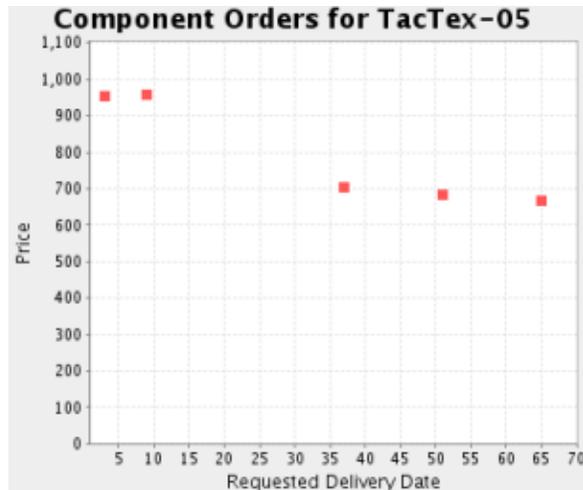


Figure 2: Component Lead Times Tool

Lead Time tool to determine what role lead time strategies played in component prices during that period.

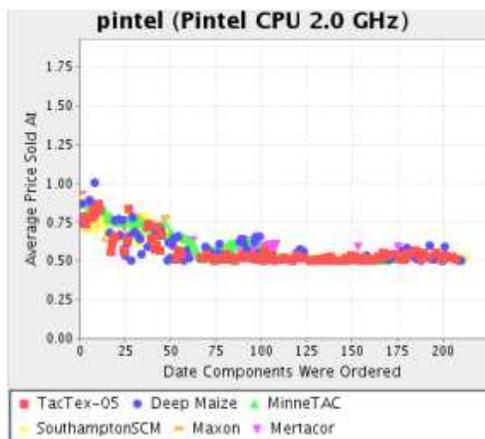
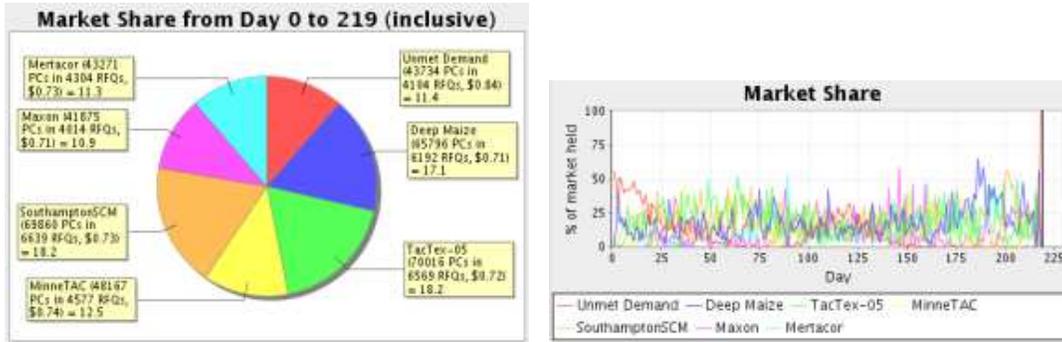


Figure 3: Component Sales Tool

- **Charting Market Shares:** the Market Share tool displays a market share pie-chart for any combination of the different customer markets captured by each agent (described in Section 6.2, shown in Figure 4). The Market Share tool allows users to change the window in which market share data is reported from any start point to any end point in the game.



(a) Market Share Pie Chart

(b) Market Share Graph

Figure 4: Market Share Tool

- Graphing Agent Bids:** users can produce graphs of the average daily winning and losing bid prices of all agents over the entire game by product type using the PC Sales tool (described in Section 6.1, shown in Figure 5). Prices can be displayed as fractions of base prices (a relative measure of quality), or dollars for different kinds of analysis. The agent bids can be correlated with changes in market share using the Market Share tool.

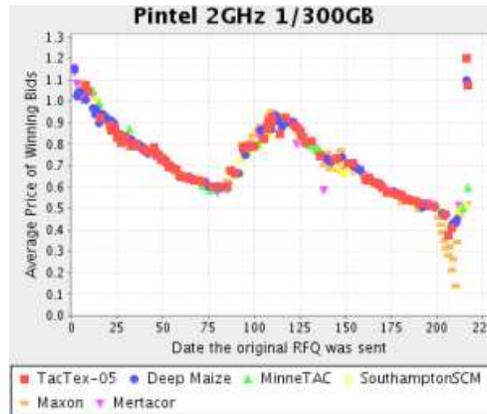


Figure 5: PC Sales Tool

3 Getting Started

3.1 Requirements

The TAC SCM Analysis and Instrumentation Toolkit was written using Java 1.5 (occasionally referred to as Java 2), and should run on any hardware capable of running the Java 1.5 virtual machine. However, it is strongly recommended having at least 128Mb of RAM available since data structures can occasionally get rather large when parsing game files.

3.2 Installation

The first step of installation is downloading the zip file containing the software. The latest version can be found at the CMU eSCM lab website, <http://www.cs.cmu.edu/~sadeh/supplychain.htm>, follow the links for downloading the TAC SCM AIT. Once you have downloaded the zip archive you should proceed to unzip its contents to the directory where you would like to store the software and game files it downloads (these game files tend to be around 2Mb each, but can be removed after use).

3.3 Running

To run the toolkit you should navigate to the directory containing the software and run the Java Virtual Machine (JVM) on the file `instrumentation.jar`, with the following command:

```
[prompt]% java -jar instrumentation.jar
```

Note: Parsing more than one game at a time may require you to allocate additional heap space to the JVM.

To expand the memory allocated to the JVM heap you can simply add the `-Xmx128m` flag to the command above, which requests that the operating system allocate 128Mb (the default is typically 64Mb) to the Java heap.

Once you have invoked the JVM you will see the main GUI window. From this window you will be able to download, parse, and analyze game files with a variety of options described later in this document.

3.4 Support

Bug reports and concerns can be sent to:

Michael Benisch

Email: mbenisch@cs.cmu.edu

3.5 Citations

We encourage agent developers and others to use our software to perform analysis for research purposes. Users are requested to acknowledge the use of these tools in any publications or presentations with the following citation:

Benisch, *et. al.*, “CMieux Analysis and Instrumentation Toolkit for TAC SCM”, Carnegie Mellon University School of Computer Science Technical Report CMU-ISRI-05-127, 2005.

4 Parsing Game Log Files

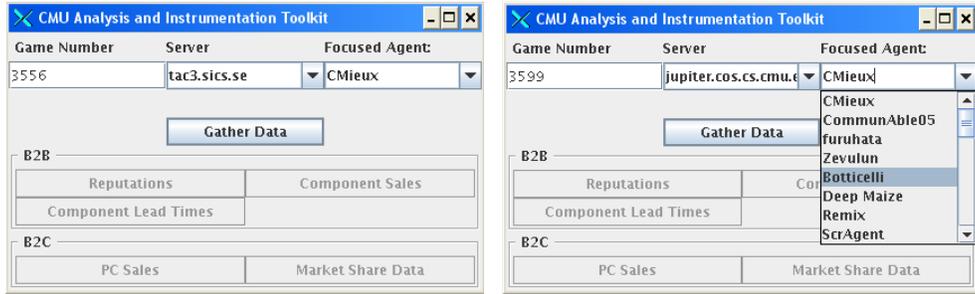
4.1 Main GUI Window

After running the executable Java archive, you will be presented with the Main GUI Window (shown in Figure 6), where you can specify the game number, server, and agent you would like to focus your analysis on (as shown in Figure 6(b)). The file `default.conf` included in the zip archive also allows you to specify the default settings of the game, server, and agent fields on the Main GUI Window for easy repeated analysis. The file has three definition lines (comments can be inserted in typical java fashion) which specify the game, server and agent as follows:

```
//default.conf; Default Configuration for CMieux Instrumentation Tool
gameNumber = 3556 // Game Number field
serverName = "tac3.sics.se" // Server field
agentName = "CMieux" // Focused Agent field
```

The fields in the Main GUI Window have the following meanings:

- **Game Number:** the number of the game you wish to analyze as specified by the TAC SCM server.
- **Server:** the web address of the server that contains the log file for the game you wish to analyze, this address must be valid and must be running a copy of the TAC SCM server (“localhost” is an acceptable address if the server is running on the local machine).
- **Focused Agent:** the agent that you would like to focus your analysis on. If the agent played in the specified game, this agent will displayed in an easily readable **RED** color on most graphs, and its data series will appear in front of others (if the agent did not play in the game this field will be ignored).



(a) Main GUI Window

(b) Specifying Game, Server, and Agent

Figure 6: Main GUI Window Interactions

4.2 Gathering Data

After specifying the relevant information in the Main GUI Window, you can press the “Gather Data” button to begin the gathering process. This process has two phases, the first phase attempts to locate the log file on the local machine in the default storage path,

```
[ait_directory]/games/[server]/[game_number]/game.slg.gz
```

if this file does not exist in this location the tool attempts to download the file from the specified server and stores it locally in that path. This allows repeated analysis of a particular game without re-downloading each time (if disk space is a concern you may remove the games directory after each usage, but you will be forced to re-download the log files).

After the file has been locally archived and located, the second phase begins, which involves parsing the file to extract relevant analysis data. If the log file is not well formed according to the SICS 2005 TAC SCM standard, this phase will throw a Java exception and you may be forced to re-start the software. Once the file is finished parsing (a progress bar will be displayed in the terminal), the analysis buttons will become enabled. Invoking a particular analysis tool is as simple as clicking the associated button, you can have as many tools open at once as you wish. The individual tools are described in more detail in later sections.

5 B2B Instrumentation and Analysis Tools

The Business to Business tools allow users to analyze the interactions between TAC SCM agents and the simulated suppliers. There are currently three tools of this nature, a reputation tool, a sales tool, and a lead time tool.

5.1 Reputation Tool

The Reputation Tool provides six graphs of agent reputations, one for each agent that competed in the specified game. An example graph is shown in Figure 1, each of the suppliers is graphed in a different color (mapped in the key below the graph), the Y-axis increases in reputation, and the X-axis increases in game day. All agents begin with a reputation of 1.0 and should have a point in the top left corner of every graph. The reputations specified are taken directly from the server log, they are the exact reputations used by the supplier on the day specified, and have been calculated by the server according to the TAC SCM Specification (including forgiveness).

5.2 Component Sales Tool

The Component Sales Tool allows users to analyze the prices at which agents purchased components from suppliers throughout the game. The tabs at the top of the window allow users to focus on different markets such as processors, motherboards, or memory, and the tabs labeled “mixed” display samples from each market together. A sample graph from a single component is shown in Figure 3. The overall weighted game averages for each agent for a particular component can be found just below the graph for that component in tool display window. Each of points along the X-axis in the graph represents a single day in the simulation, and the Y-axis value indicates the average price, *weighted by quantity*, the agent paid for the component in question on the day indicated by the X value. The prices can be changed from fraction of component base price to dollar amounts, and agents can be removed or added to the graphs using the UI controls at the bottom of the tool window.

Note: If an agent was specified for focus during the data gathering stage, the data produced by that agent is displayed in **RED** and placed in front of others on all of the sales graphs.

Note: Hovering the mouse cursor over a particular point provides a tool-tip with useful information about that point.

5.3 Component Lead Times Tool

This tool is designed to provide users with a graphical view of the lead times used by all the agents over the course of the game, as well as the prices these lead times resulted in. An example graph of lead times and prices is shown in Figure 2, the points represent individual orders placed with the specified supplier for the specified component on the specified day. The supplier and component can be specified with the drop down menu at the bottom left corner of the tool, the day can be specified with the slider at the bottom center of the tool, and various display options can be specified with the options boxes in the bottom right corner of the tool.

By default the X-axis indicates the lead time of the order, increasing in days from left to right, and the Y-axis displays the unit price of the order in dollars. All graphs have a default X-axis range of 0-50 days in the future, but will automatically expand when necessary, and a default Y-axis range of 0- H_G , where H_G is the highest price paid for the specified component over the course

of the entire game (this prevents rescaling of the Y-axis as the slider is moved). The options in the bottom right corner allow the user to change the meaning of the Y-axis from the unit price of the order to the quantity of the order, and change the Y-axis range from $0 - H_G$ to $0 - H_d$, where H_d is the highest price paid (or quantity requested) for the component that day.

Note: Hovering the mouse cursor over a particular point provides a tool-tip with useful information about that point, including reserve price, unit price, quantity, and lead time.

6 B2C Instrumentation and Analysis Tools

The Business to Customer tools focus on agent bids for customer requests. These two tools help users analyze the bidding performance of particular agents on each day in the game.

6.1 PC Sales Tool

The PC Sales tool is designed to help users analyze agents' selling prices throughout the course of the game across different types of PCs. The tool is separated into five tabbed displays, each tab corresponds to a different PC market (low, middle, high, and two mixed markets). In each display, the sales of each of the PC types that compose the corresponding market are presented as graphs (shown in Figure 5). The X-axis of the graphs increase in days, the right most point corresponding to the end of the game. The Y-axis increases in the average winning bid price for the specified PC type. There are a series of points for each agent (mapped in the key below the graph), and each point represents the average winning bid price for the agent on a particular day for the specified PC type. Below the graphs there are boxes that summarize average prices for the corresponding PC type, across the entire game for each agent. All of the averages reported by this tool are weighted by RFQ, not by the quantity of the RFQs. For example, winning an RFQ for 1 PC at \$1,000, and an RFQ for 10 PCs at \$500 would result in an average sales price of \$750, not \$545. We chose this weighting scheme because we believed it provided better insight into agent performance, in the future the option to switch weighting schemes may be added.

At the bottom of the tool there are options allowing users to customize the graphs in different ways. There are several boxes which can be checked or un-checked to remove agents' series from the graphs. There is an option to display prices in dollars or fractions of base prices, and there is an option to display average agent *losing* bid prices, rather than winning bid prices.

Note: If an agent was specified for focus during the data gathering stage, the data produced by that agent is displayed in **RED** and placed in front of others on all of the sales graphs.

Note: Hovering the mouse cursor over a particular point provides a tool-tip with useful information about that point.

6.2 Market Share Tool

The Market Share tool allows users to analyze arbitrary combinations of the different TAC SCM customer markets during different simulation periods. The Market Share tool consists of two information displays, the market share pie chart (shown in Figure 4(a)) in the left center section, and the market share graph (shown in Figure 4(b)) in the top left corner. The pie chart indicates the percentage of the selected market each agent received orders for during the specified time period, as well as the percentage of the demand that was unmet. The tips extending from the pie chart pieces provide the exact numeric percentage of the market the agent captured, the average price (as a fraction of base) of the orders for the corresponding agent in the specified market and time period, the number of PCs and the number of RFQs the agent captured.

The graph provides an overview of the market shares of each agent over the entire game for the specified market. The X-axis increases with day in the game, and the Y-axis increases with market share. Each agent is represented by a different colored line, and can be removed from the graph with the check boxes below the graph. Vertical black lines also indicate the beginning and end of the time frame selected with the UI sliders.

The rest of the tool is composed of options that allow users to customize the data that is presented in the displays. Several check boxes allow users to select different markets to include or remove from the analysis. For example, by default all check boxes are set to the on setting, and users can focus only on the low reserve RFQs (RFQs with reserve prices in the lower third of their possible reserve prices) by setting the high and mid reserve boxes to off.

In this way any combination of markets can be analyzed and both of the information displays will be adjusted to reflect the changes. Finally there are two sliders that allow users to adjust the lower and upper bound days over which the analysis spans. Both of these bounds are inclusive, and by default set to span the entire game. Users can focus the analysis on a particular segment of the game by adjusting these bounds and will notice the changes reflected in the information displays as they are changed.

7 Sample Game Analysis

In this section we will step through an example of the kind of analysis that can be performed using the AIT on a game from the 2005 TAC SCM Finals. The 2005 Finals consisted of 16 games spread evenly across two servers, the scores of the games were averaged and the agent with the highest average was deemed the winner. The first place agent, TacTex-05, was developed by the University of Texas Austin, and the second place agent, SouthamptonSCM, was developed by Southampton University. We have selected one representative game from the Finals, `3721@tac3.sics.se`, to use in our sample analysis with the AIT. After a game ends, the simulation server reports statistics such as final score, factory utilization, total material costs, and total revenue. In this game first place TacTex-05 had a better final score than second place SouthamptonSCM by about \$4 million, and it is not clear from the simple statistics reported by the server why TacTex-05 ended the game ahead. Instead, we will use the tools described in this document to determine how TacTex-05 gained its edge in game `3721@tac3.sics.se`.

The first thing we will do is enter the game number, 3721, and server, `tac3.sics.se`, into the Main GUI Window. Then we will focus our analysis on the winning agent, TacTex-05, and begin the gathering process. Once the gathering is complete we can begin our analysis by looking at the PC Sales tool, where at a glance we can see if there was a major difference in the winning bid prices of the two agents on any of the PC products. We can see that in this game the winning bid price graphs show no discernible difference between the performance of the two agents, certainly not enough to account for the \$4 million difference.

We can confirm the similar bidding results from the Market Share tool by examining overall market share information of the two agents in question. We can see that in game 3721 the two agents had nearly the same share of the entire market, about 18%, and won this share at nearly the same fraction of component base price, about 0.72. This suggests that the difference in final score was a result of the component costs of the particular markets the agents focused on. Rather than searching through the different possible markets on the Market Share tool to determine how this happened, we can switch to the Component Sales tool to determine which of the components was responsible for the difference in performance.

By examining the overall weighted average component sale prices of the different processors we begin to understand where the difference in agent performance stemmed from. We can see that TacTex-05 focused much of its attention on the Pintel 2.0 Ghz processors (a component in many of the low-end market PCs), procuring nearly 1.5 times as many of these processors as SouthamptonSCM, but paying about 7% less per unit. SouthamptonSCM seemed to have focused on the IMD 5.0Ghz processors (a component in many of the high-end market PCs), procuring about 1.3 times as many as TacTex-05, but paying only 3% less per unit. The difference between TacTex-05's focus on low end PC components, and SouthamptonSCM's focus on high end PC components saved them about 4% on materials in a game where total material costs ended around \$100 million, is likely the main reason for the \$4 million difference in final scores.

If we wanted to know how it was that TacTex-05 was able to procure its low-end processors so much cheaper than SouthamptonSCM we can turn to the Component Lead Time tool. We will select the component we have just identified as giving TacTex-05 its edge, the Pintel 2.0 Ghz processor, to focus our lead time analysis. By setting the slider to day 1, the first day in the simulation that orders were placed, we can see that right from the start TacTex-05 had a large advantage from its first set of lead times. TacTex-05 opted to send a few requests at the very start of the game with very long lead times, whereas SouthamptonSCM, in fact nearly all of the other agents, requested only short lead times on the very first day. These long-lead time requests from TacTex-05 had very large quantities, and since no other agents had similar lead times on their requests, were given prices about 3-4% less. As we slide the day slider forward in the game, however, we see that TacTex-05 abandoned its long lead time requests and decided to limit them to short lead times. Whereas SouthamptonSCM and others added one or more requests with a long lead time that was consistently 1-2% more expensive than all of TacTex-05's orders. This gives us a good picture of how, in game 3721@`tac3.sics.se` TacTex-05 was able to adapt its lead time strategy and focus on a market with a higher profit margin to end ahead of its closest competitor by about \$4 million.

References

- [1] ARUNACHALAM, R., AND SADEH, N. The supply chain trading agent competition. In *Electronic Commerce Research Application (ECRA)* (2005), vol. 4, pp. 63–81.
- [2] COLLINS, J., ARUNACHALAM, R., SADEH, N., ERIKSSON, J., FINNE, N., AND JANSON, S. The supply chain management game for 2005 trading agent competition, 2005.
- [3] SADEH, N., ARUNACHALAM, R., ERIKSSON, J., FINNE, N., AND JANSON, S. TAC-03: A supply-chain trading competition. In *AI Magazine* (2003), vol. 24, pp. 92–94.