Coordinating Decisions in a Supply-Chain Trading Agent

Wolfgang Ketter, John Collins†, and Maria Gini†⋆

Dept of DIS, RSM Erasmus University, Rotterdam, NL
†Dept of CSE, University of Minnesota, Minneapolis, MN, USA

Abstract. An autonomous trading agent is a complex piece of software that must operate in a competitive economic environment. We identify the problem of decision coordination as a crucial element in the design of an agent for TAC SCM, and we review the published literature on agent design to discover a wide variety of approaches to this problem. We believe that the existence of such variety is an indication that much is yet to be learned about designing such agents.

1 Introduction

Supply-Chain Management is an especially challenging domain for a rational decision-maker. Such an agent must not only operate simultaneously in multiple markets (a customer market and a supplier market), but it must coordinate its market activities with each other and with internal processes such as production scheduling and inventory management in a way that maximizes its utility across an extended time horizon.

Organized competitions can be an effective way to drive research and understanding in complex domains, free of the complexities and risks of operating in open, real-world environments. Artificial economic environments typically abstract certain interesting features of the real world, such as markets and competitors, demand-based prices and cost of capital, and omit others, such as human resources, secondary markets, taxes, and seasonal demand. The Trading Agent Competition for Supply-Chain Management [1] (TAC SCM) is based on an economic simulation in which competing autonomous agents operate in a simple supply-chain scenario, purchasing components, managing a factory and warehouse, and selling finished products to customers.

TAC SCM has been an active competition since 2003, and the design of the game has been stable since 2005. More than 50 different teams have participated, and a number of papers have been published that describe agent designs, agent and game analyses, and specific methods for modeling the markets and decision processes in the simulation.

TAC SCM is an interesting challenge for a number of reasons. Different groups have approached the problem from a variety of perspectives, depending on the individual interests and backgrounds of the participants. For example, a team that is primarily interested in developing and testing machine-learning techniques will have a very different approach to the problem than a team that is primarily interested in developing methods to solve constrained optimization problems under uncertainty. To better understand this variety, we conducted an informal survey of many of the active teams in

⋆ Supported in part by the National Science Foundation under award NSF/IIS-0414466.

2007. In this paper, we explore in some depth and attempt to classify the variety of approaches we have observed to one of the special challenges in designing a successful agent for TAC SCM, the problem of coordinating the various decision processes.

2 Overview of the TAC SCM game

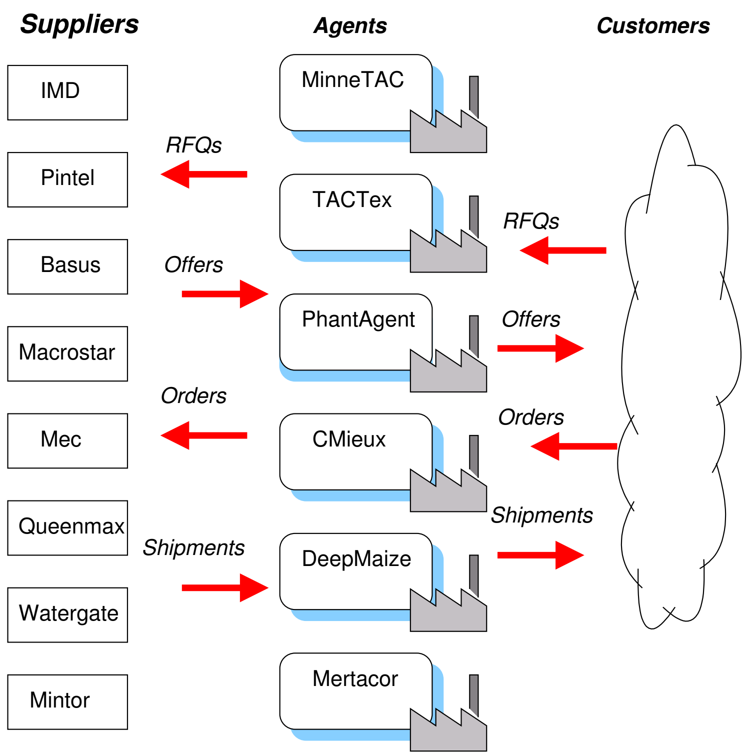

Fig. 1. TAC SCM game scenario.

In a TAC SCM game, each of the competing agents plays the part of a manufacturer of personal computers. Figure 1 gives a schematic overview of the TAC SCM game. Agents compete with each other in a procurement market for computer components, and in a sales market for customers. A game runs for 220 simulated days over about an hour of real time. Each agent starts with no inventory and an empty bank account. The agent with the largest bank account at the end of the game is the winner.

Customers express demand each day by issuing a set of Request for Quotes (RFQs) for finished computers. Each RFQ specifies the type of computer, a quantity, a due date, a reserve price, and a penalty for late delivery. Each agent may choose to bid on any subset of the day's RFQs. For each RFQ, the bid with the lowest price will be accepted, as long as that price is at or below the customer's reserve price. Once a bid is accepted, the agent is obligated to ship the requested products by the due date, or it must pay the stated penalty for each day the shipment is late. Agents do not see the bids of other agents, but aggregate market statistics are supplied to the agents periodically. The customer market is segmented into a low-cost segment with five products, a mid-range segment with six products, and a premium segment with five products. Customer demand in each segment varies independently through the course of the game by a random walk with a superimposed Poisson distribution.

2.1 Agent decision processes

Agents assemble computers from parts, which must be purchased from suppliers. When agents wish to procure parts, they issue RFQs to individual suppliers, and suppliers respond with bids that specify price and availability. If the agent decides to accept a supplier's offer, then the supplier will ship the ordered parts on or after the due date. Late shipments are possible because supplier capacity varies from day to day by a mean-reverting random walk. Supplier prices are based on the ratio of demand to current uncommitted capacity, so agents have to decide when to place their orders, for what amounts, what due dates, and at what minimum price.

Once an agent has the necessary parts to assemble computers, it must schedule production in its finite-capacity production facility. Each computer model requires a specific set of parts, and a specified number of assembly cycles. Assembled computers are added to the agent's finished-goods inventory, and may be shipped to customers to satisfy outstanding orders.

An agent operating in the TAC SCM scenario must make the following four basic decisions during each simulated "day" in a competition:

1. decide what parts to purchase, from whom, and when to have them delivered (Procurement).
2. schedule its manufacturing facility (Production).
3. decide which customer RFQs to respond to, and set bid prices (Sales).
4. ship completed orders to customers (Fulfillment).

These decisions are supported by models of the sales and procurement markets, and by models of the agent's own production facility and inventory situation. The details of these models and decision processes are the primary subjects of research for participants in TAC SCM. These models may be populated with historical data from previous games, and with observations in the current game. During a game, agents can observe market reactions to their own actions (bids accepted or not, price and quantity data in supplier offers), and a very limited set of market summary data. In-game summary information is limited to daily high and low order prices for each product in the customer market, and summary reports every 20 days that give average prices and aggregate quantities. Many important factors, such as current capacity and outstanding commitments of suppliers, and sales volumes and price distributions in the customer market, are not visible to the agents.

2.2 Game balance

The design of TAC SCM was carefully tuned over the first three years to make the competition interesting and challenging. The most obvious opportunities for strategic manipulation [2, 3] have been eliminated. Agents must manage their reputations with respect to each supplier, to discourage agents from making large requests and then turning down the resulting offers. Suppliers reserve approximately half of their total capacity at the beginning of the game for future demand, which makes it very difficult to "corner" the market for some component type.

The parameters of the game scenario are set to ensure that decision coordination among procurement and sales is reasonably challenging. Figure 2 shows the overall balance between supply and demand. It is a histogram of the daily customer RFQ count over 200 games, about 40,000 observations. Superimposed on the histogram are the mean customer demand, the aggregate capacity of the six agent factories, and the expected supplier capacity. The key message from this balance is that expected customer demand is somewhat below the expected ability of the market to supply that demand. This means that an agent can expect to buy enough parts to keep its factory busy, but a strategy that simply tries to keep the factory busy all the time is likely to result in a large unsold inventory at the end of an average game[1]. On the other hand, there are some games in which the agents cannot supply all the demand, and the variability of the game can lead to serious imbalances between customer demand for specific products and the availability of parts to build them.

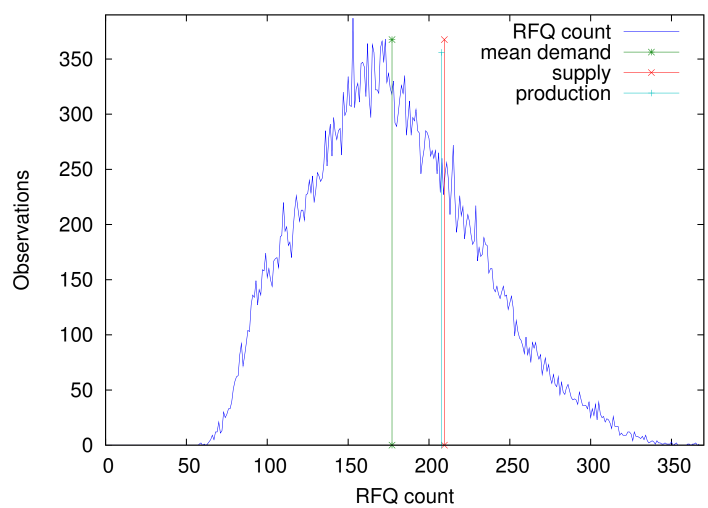

Fig. 2. Game balance. Mean customer demand is below the production capacity of all the agents, and below the expected availability of parts in the supplier market.

[1] This balance was first introduced in the 2005 competition. Price wars were a large problem in the early rounds of that competition until the full-production agents were eliminated

3 Agent design and the decision coordination problem

Kiekintveld et al. [4] identify three key issues that a successful TAC SCM agent must address: dealing with substantial *uncertainty* in a highly *dynamic* economic environment, in competition with other self-interested agents whose behavior is naturally *strategic*. To this list we would add two other issues: making *coordinated* decisions across multiple domains in order to maximize payoff over time, and operating effectively in an *oligopoly* market. Because of the relatively small number of players in the customer and supplier markets, both are best characterized as oligopoly markets, and so many of the simplifying assumptions that can be used effectively in large markets do not hold. One such assumption is that the decisions of individual players have negligible impact on the observable market behavior. In contrast, competitive TAC SCM agents cannot act simply as "price-takers" in these markets - their decisions can move the markets decisively, and failure to account for this reaction can cause significant deviations from predicted outcomes.

The complexity of the problems an agent must solve to be competitive in TAC SCM has produced a number of interesting design approaches. To understand the spectrum of agent designs, we conducted a survey of the research community via the TAC SCM discussion email list in the period May-September 2007. Some common themes were an emphasis on modularity, use of constrained optimization techniques, machine learning, dealing with uncertainty, and a focus on coordination of decisions among procurement, production, sales, and fulfilment. Detailed results of the survey are presented in [5].

Two of the four agent decision problems, procurement and sales, are dominated by the variability of the game scenario and are strongly affected by the actions of other agents, while the production-scheduling and fulfillment decisions are internal to the agent and less affected by the inherent variability in the game. Because of this, some agent designs simply fold fulfillment into the sales problem, and production scheduling is sometimes also bundled into sales, especially for agents that use a make-to-order production strategy.

Simply stated, a solution to coordination problem will maximize (expected) profit over an entire game, subject to availability of individual part types in the supplier market, demand in the customer market, and capacity of the agent's factory. Of course, prices and availability in the supplier market are at least partly determined by the behavior of other agents in the simulation. In addition, prices in the customer market are largely determined by the behavior of the other agents, since competition almost always keeps prices well below customer reserve prices. This problem is commonly viewed as one of enabling independent decision processes to coordinate their actions while minimizing the need to share representation and implementation details.

As we shall see, many approaches to the coordination problem have been tried, and there is little evidence from tournament standings that any of these approaches dominates the others. In fact, a study by Jordan et al. [6] has shown that no single dominant strategy has yet been found, and our analysis shows that the top three agents in the Jordan study, namely TacTex, DeepMaize, and PhantAgent, use different coordination mechanisms. We do know that the "push" strategy that was popular in the 2003 and 2004 competitions (for example, Benisch et al. [7]) is not effective, because the factory can produce more than what can be sold at a profit, at least in expectation. This approach attempts to purchase enough parts early in the game to keep factory utilization high for the entire game, thereby eliminating procurement from the coordination problem.

In the following sections, we explore the variety of coordination approaches that we have observed among published agent designs and the respondents to the 2007 survey. We note that none of them have tried to solve the general problem in its entirety, presumably because the variability inherent in the simulation and the difficulty of predicting the behaviors of other agents have so far defeated all attempts to do so. Therefore, what we see is that each design has chosen a more manageable approach, one that simplifies the problem through approximations, through heuristics, and through focus on much shorter time horizons than the entire game.

3.1 Predicted sales volume

Because the balance of supply, demand, and production capacity in the simulation design has defeated a simple “push” approach to coordination, the next obvious choice would seem to be adoption of a “pull” approach, in which sales activities pull finished goods through the factory, which in turn pulls in components through the procurement market.

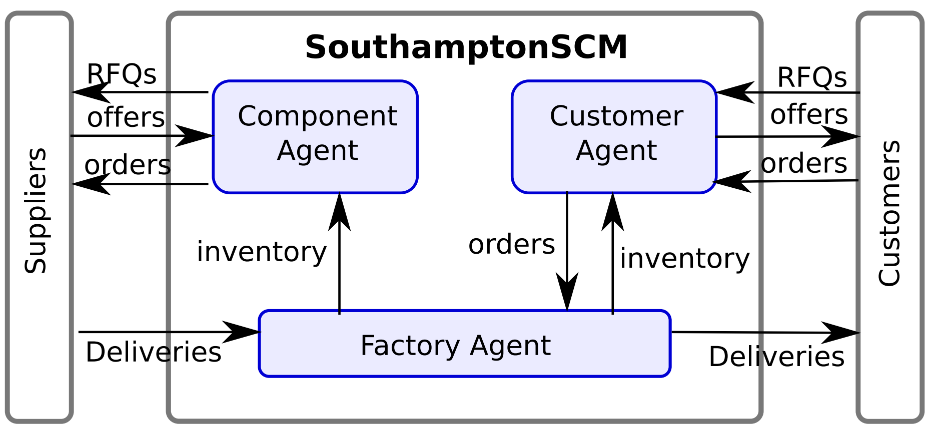

Fig. 3. SouthamptonSCM: Coordination primarily by the Customer Agent.

A good example of this approach is SouthamptonSCM [8]. This agent was a finalist in the 2004 competition, and placed second in 2005. Figure 3 is a schematic representation of this design. In SouthamptonSCM, a Customer Agent uses fuzzy reasoning to compute offer prices, based on inventory level, customer demand, and time in the game[2]. Priority is given to the products with the highest expected per-unit profit. The Component Agent buys a portion of its components with long lead-times, because

[2] A separate rule set is used near the end of the game, because of the need to exhaust inventory and because prices tend to be much more volatile late in the game

prices tend to be lower with longer lead times. The remaining component inventory is purchased with shorter lead times, in response to observed customer demand and to depletion of inventory by sales to customers. The Factory Agent primarily builds outstanding customer orders, and if it has spare capacity and available parts, it builds up a modest inventory of finished goods.

The MinneTAC agent [9, 10], shown in Figure 4, can be configured in a number of different ways, but the configuration used in the 2007 and 2008 competitions solves a linear program each day to maximize expected profit over a 20-day horizon, subject to constraints on production capacity, customer demand, and anticipated inventory. The output of the linear program is "sales quotas" for each product for each of the next 20 days. The current-day quota is used by the Sales Manager to set prices in the customer market, and future-day quotas are used by the Supplier Manager to drive procurement. A slightly different configuration of MinneTAC was a finalist in the 2005 and 2006 competitions.

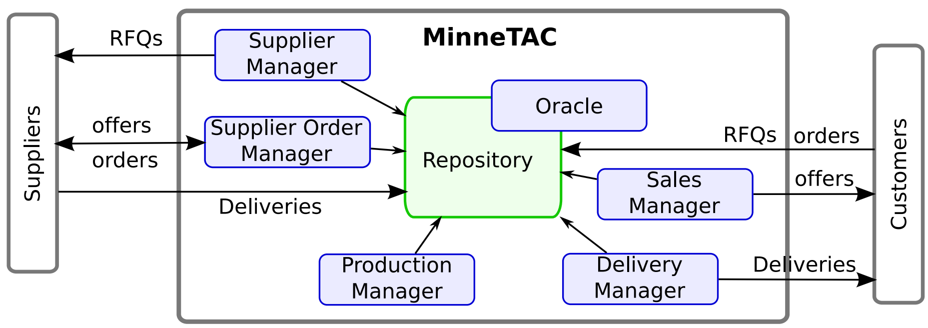

Fig. 4. MinneTAC: Coordination through the Repository, details depend on configuration.

As we can see from Figure 4, MinneTAC uses a very different design approach from the other agents we examine here. The Repository acts as a "blackboard", and the various components interact only through the Repository. The Oracle component is a wrapper for a large number of small modules, called "Evaluators", that can be strung together as specified in a configuration file to do the necessary analysis and prediction tasks requested by the decision components. The actual coordination among decision components happens because they share some of those Evaluators. Specifically, both the Sales Manager and the Supplier Manager use the sales quotas produced by one of the Evaluators.

3.2 Future production schedule

DeepMaize [4] coordinates its decisions through a principled approach called "value-based decomposition". In this approach, a long-term production schedule is constructed by incrementally adding the products that are expected to return the highest profit. The general scheme is shown schematically in Figure 5.

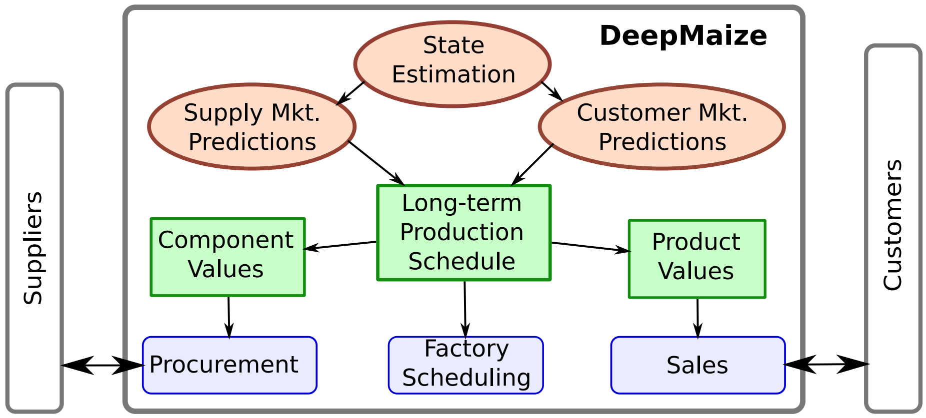

Fig. 5. DeepMaize: Coordination through a long-term production schedule, using value-based decomposition.

This approach depends on pricing models in both the customer and supplier markets that effectively capture price-quantity tradeoffs. The two prediction components shown in the diagram, along with an off-line machine-learning process, are responsible for producing those models. Given a long-term production schedule, the Procurement module attempts to provide the necessary components to fill it, and Sales uses it to set prices in the customer market. DeepMaize has been a finalist in all of the TAC SCM tournaments. It placed third in 2006 and 2007, and first in 2008.

3.3 Inventory management

Three published agent designs appear to focus on an inventory model to coordinate decisions. Mertacor [11, 12] is the clearest example. As we see in Figure 6, an "Inventory Manager" component is the central element in this design. Mertacor uses an "Assemble to Order" approach, which is recommended in the literature on inventory management for situations where assembly times are significantly shorter than procurement lead times. The Inventory Manager attempts to maintain component stocks above a minimum threshold, subject to committed and expected sales, and to committed deliveries from suppliers. Mertacor placed third in the 2005 competition.

PhantAgent [13] is another design that appears to focus on inventory management, although as we see in Figure 7, the inventory management function is conceptually combined with the procurement function in a Component Module. The goal of the Component Module is to maintain expected stocks of each component type within narrow bounds throughout the game. It computes expected stocks for each component for each day until the end of the game, and formulates new supplier orders to make up any deficits. PhantAgent placed second in 2006, and first in the 2007 competition.

For each component c, and for each day d from the current day until the end of the game, the expected stock is computed as

$$I_{c,d} = I_{c,d-1} + incoming_{c,d} - usage_{c,d}$$

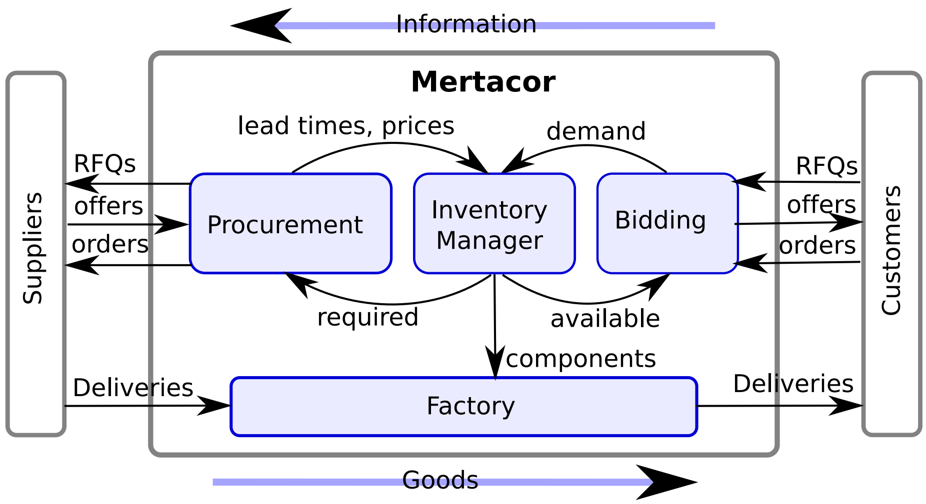

Fig. 6. Mertacor: Coordination through the Inventory Manager.

where $I_{c,d}$ is the expected inventory of component c on day d, $incoming_{c,d}$ is the quantities of committed supplier orders, and $usage_{c,d}$ is the expected usage of component c on day d.

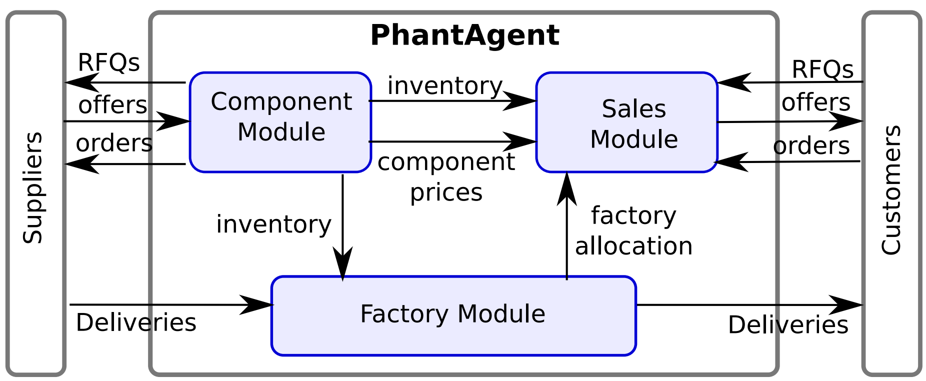

Fig. 7. PhantAgent: Coordination is by inventory control, originating in the Procurement module.

PhantAgent is interesting in another way. It deals with the inherent complexity and uncertainty of the TAC SCM environment, and the resulting strategic inter-dependencies between the different agent modules, using heuristic approximations rather than optimization algorithms. The team's assessment is that finding optimal solutions to the different sub-problems does not always lead to the best overall performance.

Another agent whose decision coordination mechanism seems focused on inventory control is CrocodileAgent [14, 15]. The agent drives procurement to maintain expected component inventory stocks within defined minimum and maximum bounds. Similarly, Production operates to maintain a finished goods inventory within pre-defined bounds. Sales then bids on customer requests using a simple pricing algorithm, in an attempt to sell products, profitably, as fast as they are being produced. When demand is low, the profitability constraint causes inventory to back up, and production and procurement to slow down.

3.4 Central strategy module

An agent that has very clearly separated the decision coordination issues from the details of procurement, sales, and production scheduling is CMieux [16], a finalist in the 2007 and 2008 competitions. A schematic diagram of the CMieux design is shown in Figure 8.

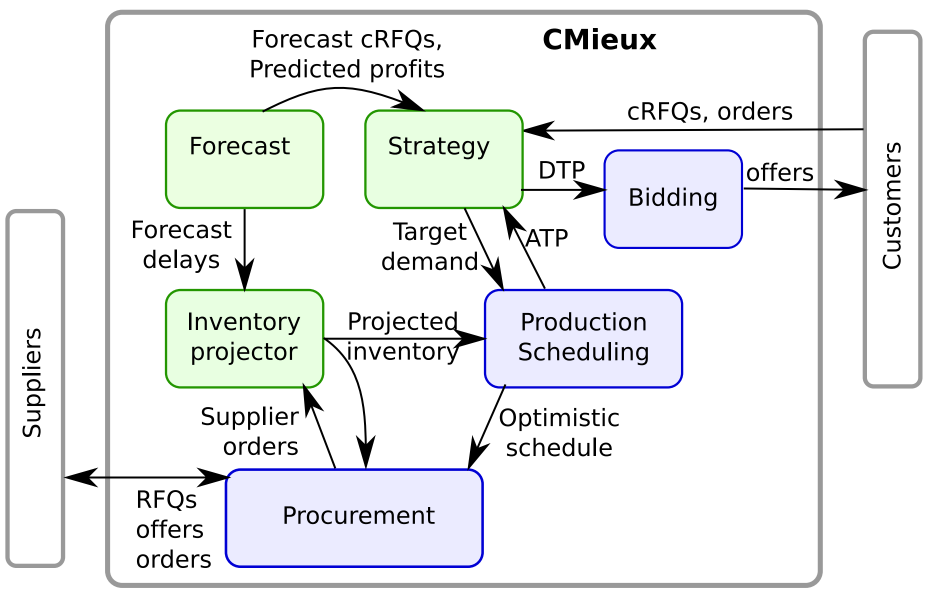

Fig. 8. CMieux: Coordination by a separate Strategy module.

The Strategy module sets overall goals for the remainder of the system, such as the portion of expected demand to target, and the portion of the production schedule (ATP, the products Available to Promise) that should be sold to customers (DTP, products Desired to Promise). The Forecast module observes the markets and makes predictions about demand, prices, and delays in supplier shipments. The Inventory Projector combines that with current inventories and expected supplier deliveries to generate inventory projections over time. Procurement uses the projected inventory along with an optimistic version of the production schedule (what Production would expect to build if there were no inventory constraints) to decide what to order from suppliers, and supplies Inventory Projector with actual supplier orders.

3.5 Separate supply and demand models

The design of TacTex [17] is quite different from the others we have reviewed, in the sense that it does not try to centralize decision coordination at all. Instead, it employs a Supply Manager that interacts with suppliers and models the supply market, and a Demand manager that interacts with customers and models the customer market. Coordination is achieved by communication between these two models. TacTex has been a very strong competitor, placing first in 2005 and 2006, and second in 2007.

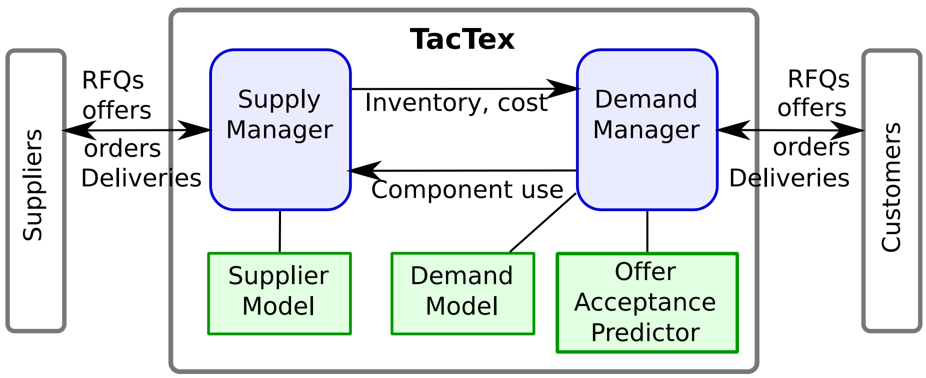

Fig. 9. TacTex: Coordination is by communication of inventory, cost, and projected usage data between the Supply Manager and the Demand Manager.

In this design, the Supply Manager attempts to minimize the cost of procuring the components requested by the Demand Manager, and provides in return an inventory projection including current inventory and expected future deliveries, along with replacement cost estimates for each component type. The Demand Manager, in turn, seeks to maximize the agent's profits from sales, subject to constraints from the customer market, its own production capacity, and the information provided by the Supply Manager.

3.6 Internal markets

RedAgent [18] is a unique approach to agent design. It won the first year's competition in 2003, but did not do well in 2004 and was never updated after the rule change in 2005.

As we can see from Figure 10, RedAgent manages the flow of components from suppliers through production and into sales and fulfillment of customer orders through

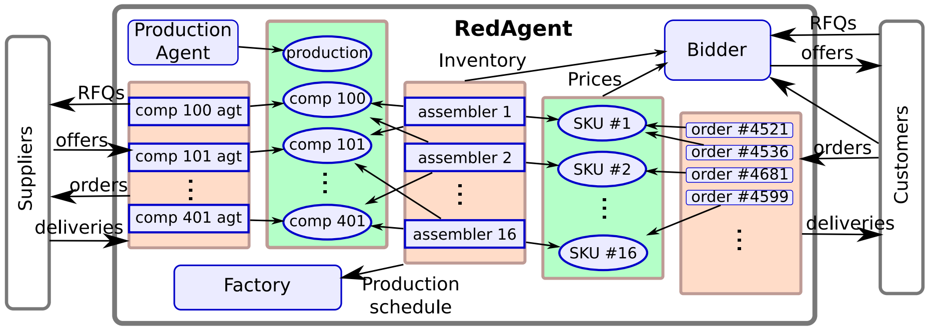

Fig. 10. RedAgent: Coordination by a sequence of internal markets.

a series of internal markets. The Bidder observes its inventory status and the current prices in its internal finished-goods markets, and makes offers to customers. Customer orders then compete for products in the internal product markets, which are supplied by assemblers for each product type. Those assemblers in turn compete for components in internal component markets, which are supplied by individual component agents. The component agents then interact with suppliers in order to set prices and supply their markets. RedAgent used loosely-coupled "sub-agents" competing with each other in internal auction-based markets for finished goods, production capacity, and components. This achieved a radical decoupling of the various components, but proved to be not competitive after the game design was adjusted in 2005 to defeat some of the simplest approaches that lacked adequate coordination among decisions. Specifically, agents that focused procurement only on keeping the factory in full production found themselves overproducing when the balance between factory capacity and expected customer demand was adjusted.

4 Conclusions and Future Work

We have presented a brief overview of agent design ideas and architectures for TAC SCM, using information both from a survey of agent development teams and from published results. The overall survey outcome shows that there are common themes emerging from the different research groups on how to design a successful supply-chain trading agent. A clear challenge that each agent design must meet is the need to coordinate its internal processes (production scheduling) with action in procurement and customer markets. We have observed a variety of approaches to the decision coordination problem, including the use of sales to "pull" products and supplies through the system, coordination through internal models of inventory and prices, assigning current and future value to inventory and production resources, and the use of an explicit top-level strategy component that coordinates the lower-level decision processes. The fact that after several years of competition there is still much to be learned, suggest that the recipe for a full competent supply-chain trading agent is still an unsolved problem, even for an abstract, constrained environment like TAC SCM.

Many of these agents are available in either binary or source form through the TAC SCM Agent Repository at http://www.sics.se/tac/showagents.php.

References

1. Collins, J., Arunachalam, R., Sadeh, N., Ericsson, J., Finne, N., Janson, S.: The Supply Chain Management Game for the 2006 Trading Agent Competition. Technical Report CMU-ISRI-05-132, Carnegie Mellon University, Pittsburgh, PA (2005)
2. Ketter, W., Kryzhnyaya, E., Damer, S., McMillen, C., Agovic, A., Collins, J., Gini, M.: Analysis and design of supply-driven strategies in TAC-SCM. In: Workshop: Trading Agent Design and Analysis at the Third Int'l Conf. on Autonomous Agents and Multi-Agent Systems, New York (2004) 44–51
3. Wellman, M.P., Estelle, J., Singh, S., Vorobeychik, Y., Kiekintveld, C., Soni, V.: Strategic interactions in a supply chain game. Computational Intelligence **21**(1) (2005) 1–26
4. Kiekintveld, C., Miller, J., Jordan, P.R., Wellman, M.P.: Controlling a Supply Chain Agent Using Value-Based Decomposition. In: Proc. of 7th ACM Conf. on Electronic Commerce, Ann Arbor, USA (2006) 208–217
5. Collins, J., Ketter, W., Gini, M.: Flexible decision control in an autonomous trading agent. Electronic Commerce Research and Applications **In Press, Corrected Proof** (2008) –
6. Jordan, P.R., Kiekintveld, C., Wellman, M.P.: Empirical game-theoretic analysis of the tac supply chain game. In: Proc. of the Sixth Int'l Conf. on Autonomous Agents and Multi-Agent Systems. (2007) 1188–1195
7. Benisch, M., Greenwald, A., Grypari, I., Lederman, R., Naroditskiy, V., Tschantz, M.: Botticelli: A supply chain management agent designed to optimize under uncertainty. ACM Trans. on Comp. Logic **4**(3) (2004) 29–37
8. He, M., Rogers, A., Luo, X., Jennings, N.R.: Designing a successful trading agent for supply chain management. In: Proc. of the Fifth Int'l Conf. on Autonomous Agents and Multi-Agent Systems. (2006) 1159–1166
9. Collins, J., Ketter, W., Gini, M.: Flexible decision support in a dynamic business network. In Verwest, P., van Liere, D., Zheng, L., eds.: The Network Experience – New Value from Smart Business Networks. Springer Verlag (2008) 233–246
10. Collins, J., Ketter, W., Gini, M., Agovic, A.: Software architecture of the MinneTAC supply-chain trading agent. Technical Report 08-031, University of Minnesota, Department of Computer Science and Engineering, Minneapolis, MN (2008)
11. Chatzidimitriou, K.C., Symeonidis, A.L., Kontogounis, I., Mitkas, P.A.: Agent Mertacor: A robust design for dealing with uncertainty and variation in SCM environments. Expert Systems with Applications **35**(3) (2008) 591–603
12. Kontogounis, I., Chatzidimitriou, K., Symeonidis, A., Mitkas, P.: A Robust Agent Design for Dynamic SCM Environments. Proceedings of the 4th Hellenic Joint Conference on Artificial Intelligence (SETN), Heraklion, Greece (2006) 127–136
13. Stan, M., Stan, B., Florea, A.M.: A Dynamic Strategy Agent for Supply Chain Management. Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing **0** (2006) 227–232
14. Podobnik, V., Petric, A., Jezic, G.: The crocodileagent: Research for efficient agent-based cross-enterprise processes. In Meersman, R., Tari, Z., Herrero, P., eds.: On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. Volume 4277. Springer-Verlag (2006) 752–762
15. Petric, A., Podobnik, V., Jezic, G.: The CrocodileAgent: Designing a robust trading agent for volatile e-market conditions. In Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C., eds.:

Agent and Multi-Agent Systems: Technologies and Applications. Volume 4496 of LNAI. Springer-Verlag (2007) 597–606
16. Benisch, M., Sardinha, A., Andrews, J., Sadeh, N.: CMieux: adaptive strategies for competitive supply chain trading. In: Proc. of 8th Int'l Conf. on Electronic Commerce, ACM Press (2006) 47–58
17. Pardoe, D., Stone, P.: An autonomous agent for supply chain management. In Adomavicius, G., Gupta, A., eds.: Handbooks in Information Systems Series: Business Computing. Elsevier (2007)
18. Keller, P.W., Duguay, F.O., Precup, D.: Redagent - winner of the TAC SCM 2003. SIGecom Exchanges **4**(3) (2004) 1–8