Flexible Decision Support in a Dynamic Business Network

John Collins, Wolfgang Ketter, and Maria Gini

Abstract We present the design of a service oriented architecture which facilitates flexible managerial decision making in dynamic business networks. We have implemented and tested this architecture in the MinneTAC trading agent, which is designed to compete in the Supply Chain Trading Agent Competition [4]. Our design enables managers to break out decision behaviors into separate, configurable components, and allows dynamic construction of analysis and modeling tools from small, single-purpose "evaluator" services. The result of our design is that the network can easily be configured to test a new theory and analyze the impact of various approaches to different elements of the agent's decision processes, such as procurement, sales, production, and inventory management. Additionally we describe visualizers that allow managers to see and manipulate the configuration of the network, and to construct economic dashboards that can display the current and historical state of any node in the network.

1 Introduction

Organizations in business networks have a growing need for intelligent software that can assist managers by gathering and analyzing information, making recommendations, and supporting business decisions. Advanced decision support systems and autonomous software agents promise to address this need by acting rationally on behalf of humans in numerous application do-

John Collins
Computer Science and Engineering, University of Minnesota, e-mail: jcollins@cs.umn.edu
Wolfgang Ketter
Decision and Information Sciences, RSM Erasmus University, e-mail: wketter@rsm.nl
Maria Gini
Computer Science and Engineering, University of Minnesota, e-mail: gini@cs.umn.edu

mains. Examples include procurement [33, 7], scheduling and resource management [16, 5], and personal information management [2]. The recent advent of *Smart Business Networks* (SBN) [37, 36, 13] extends the area of traditional business processes and gives rise to new challenges, especially in the area of dynamic and modular business process management, by providing the ability to integrate legacy systems and by providing advanced tools to facilitate human managerial decision making.

We make four major contributions to the SBN literature. One of the major theoretical tenets of SBNs is the ability of actors to quickly connect to other actors to achieve specific business objectives and then disconnect when a task is finished. Our first contribution in this paper extends the SBN literature through the design and practical implementation of a highly configurable and flexible decision support system that dynamically connects to different nodes of a business network and disconnects them when no longer needed. Our second contribution is the vision of goal directed service composition. This allows business services with formal semantic descriptions to be composed and validated. Thirdly, we are developing a tool that lets managers visualize, understand, and validate the theoretically designed decision chain with a graphical representation of the actual network chain. Finally, we have developed a flexible economic dashboard architecture that can be dynamically connected to selected nodes to visualize their real-time status, current parts and finished goods inventory positions, risk and reward management, and the like. This architecture can greatly empower business network managers in their understanding of the overall business network structure and facilitate real-time managerial decision making. Currently, we are working on an even more interactive version of this dashboard which allows the human decision maker to interact with the business network to make structural changes.

Since operating on real world business networks has high risks, and might generate serious business problems when not done properly, we tested our architecture and algorithms on a supply-chain testbed, the Trading Agent Competition for Supply Chain Management [4] (TAC SCM). We describe the implementation of our flexible decision support system and demonstrate its value using as an example MinneTAC [6], an autonomous agent that performs coordinated buying, selling, production scheduling, and inventory management in the context of TAC SCM. In addition, we present results of our network visualizer toolbox, where a manager is able to see the current configuration of the network as well as the state of the different nodes. We review the relevant related literature, and finish with conclusions and future work. In the future work section we describe the Dutch flower auction network as an example of a complex, strategic, and uncertain business network on which we are currently working to integrate our architecture and algorithms.

2 A Business Network Testbed: The Trading Agent Competition for Supply Chain Management

Traditionally, supply chains have been created and maintained through the interactions of human representatives of the various enterprises (component suppliers, manufactures, wholesalers/distributors, retailer and customers) involved. However, the recent advent of autonomous trading agents opens new possibilities for automating and coordinating the decision making processes between the various parties involved. The Trading Agent Competition for Supply Chain Management (TAC SCM) is an abstract model of a highly dynamic direct sales [3] environment, as exemplified by Dell Inc.[1], for procurement, inventory management, production, and sales.

TAC SCM simulates a product life-cycle for a manufacturing organization. In the simulation scenario, each of six competing agents plays the part of a manufacturer of personal computers. Agents compete with each other in a procurement market for computer components, and in an auction-based sales market to sell computers to customers, as shown in Figure 1. The scenario models a market situation where products have limited market life, and the major components used to manufacture those products have little or no residual value at the end of that market life. A typical simulation runs for 220 simulated days over about an hour of real time. Each agent starts with no inventory, an empty bank account, and a finite-capacity production facility. Agents must borrow (and pay interest) to build up inventory of computer components before they can begin assembling and shipping computers. Agents have very limited visibility of the actions of other agents, and must

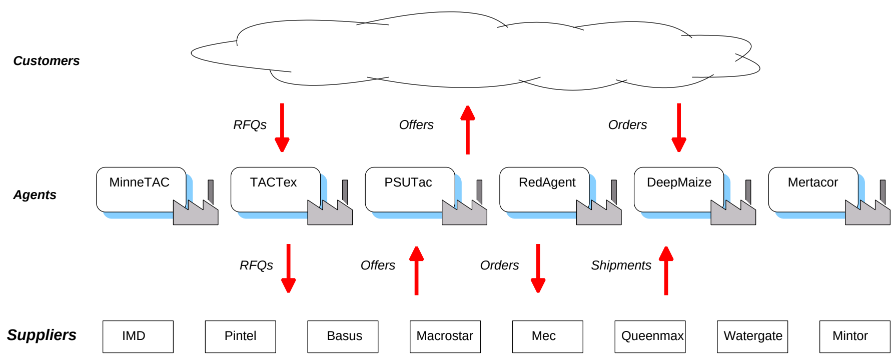

Fig. 1 Schematic overview of a typical TAC SCM game scenario. Agents submit daily Request for Quotes (RFQ) to suppliers to buy component part, and customers request finished computers.

deal with significant variability in customer demand, supplier capacities, and

[1] http://www.dell.com

other factors. The primary performance criterion is profitability, so the agent with the largest bank account at the end of the simulated year is the winner.

Organized competitions can be an effective way to drive research and improve understanding in complex domains, free of the complexities and risk 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 personalities, taxes, and seasonal demand. Examples related to electronic commerce, besides TAC SCM, include the Penn-Lehman Automated Trading Project [20], the TAC travel competition [40], and the CAT competition [28].

3 Designing an Intelligent Trading Agent for Dynamic Business Networks

Since the inception of TAC SCM in 2002, more than 50 teams have built and entered agents. These agents represent a variety of approaches to solving the various modeling and decision problems presented by the simulation scenario. We wanted our agent to be a flexible research tool, to enable easy testing of hypotheses and comparison of approaches. We intend to use MinneTAC as a teaching tool, to teach concepts in supply-chain management, economic decision making, machine learning, and software design. To address the twin challenges of simulating a business organization and supporting a research agenda, the design of MinneTAC [6] models a flexible organization using a service-oriented approach. There are a few top-level decision elements (Procurement, Manufacturing, Sales) and a large number of services that act as analysis modules, supported by a common database. We call these modules *evaluators*. A high-level schematic representation of this design is shown in Figure 2. Decision components operate by retrieving data from the database, and evaluation results from evaluators. Evaluators share a common service-oriented design, and they may be composed into chains and feedback loops to perform arbitrarily complex analyses. They may request inputs from other evaluators, from the database, and from external sources. They then transform that data in various ways, for example by updating price models, estimating demand trends, or running optimization formulas to produce sales quotas or procurement recommendations. Results are provided in a common, self-describing format so they can be used by other evaluators or decision components. Connections among decision components and evaluators are entirely configurable and modifiable at runtime; the only real dependency in this design is on the database, and on external data sources such as market data and user inputs. This allows individual researchers to encapsulate modeling and decision problems within the bounds of individual components and services that have minimal, well-defined interactions among themselves.

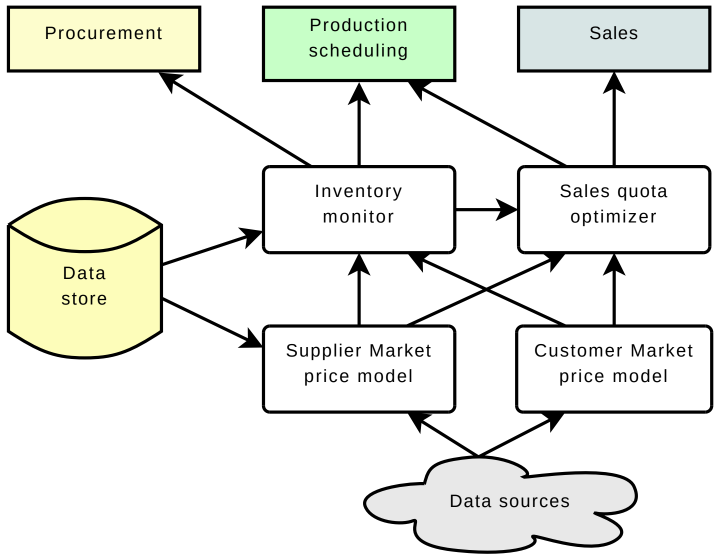

Fig. 2 MinneTAC trading agent architecture. Arrows show data flow, not dependencies.

In Figure 2, the primary decision components are shown across the top. The Procurement component deals with suppliers, attempting to find the parts needed by Manufacturing at the lowest possible cost. Manufacturing schedules the production facility with assembly tasks that maximize the expected value of its available inventory and production capacity. Sales sets prices and makes customer offers that are expected to maximize profit, given its available resources. These three decision components are in turn supported by a common data store, and by a large set of evaluators that perform various modeling, analysis, and prediction tasks. These are represented schematically here as the interconnected blocks in the center of the diagram, the "Sales Quota Optimizer," the "Customer Market Price Model," etc. The evaluators, in turn, have access to each other and to various internal and external data sources, primarily in the form of periodic market reports that are issued by the simulation, and a large body of historical data that has been "digested" by machine learning models, such as the "economic regime" model described by Ketter [23, 24].

The radical separation of the MinneTAC agent design into separate decision processes and evaluator services addresses the needs of researchers, who need short learning curves and low risk of interfering with each other. Does it serve the needs of the agent itself, which must effectively coordinate its decisions? The most obvious coordination methods are the "push" approach, in which Procurement tries to keep the factory busy and Sales works to maximize profits on the resulting finished goods, and the "pull" approach, in which manufacturing and procurement work to maintain target inventory levels at minimum cost as Sales finds profitable opportunities to sell the available inventory. Another possible approach to the coordination problem is the one used by the RedAgent team at McGill University [21], in which the primary decision components communicate through internal auction-based markets. The DeepMaize team at Michigan [25] uses a projected production schedule as the primary coordination structure. Slots in the schedule are filled with products that are expected to return the highest marginal profit at some point in the future. Procurement then works to provide sufficient inventory to run the projected schedule, and sales works to sell what is produced.

In MinneTAC, the database holds a record of all transactions made in the past, as well as inventory data, current customer requests, and supplier offers. The evaluators use this data, along with their own data sources, to produce analyses and recommendations that drive decisions. The version of MinneTAC that ran in the 2007 Trading Agent Competition used a modified "pull" method to coordinate its decisions. It was configured to use current and projected sales quotas over an extended time horizon as the primary coordination mechanism, to drive not only sales, but also production and short-term procurement. Long-term procurement was based on estimates of future customer demand, which is produced by another evaluator, and also used as an input for generating sales quotas.

Evaluators can be composed into arbitrarily complex structures, through a back-chaining process. They do this by requesting the outputs of other Evaluator services in the process of producing their results. Such Evaluation requests are made by name rather than by direct reference, and these names are configurable, either through XML configuration files, or through a user interface. This approach preserves independence among Evaluator services, and it elevates and makes visible the high-level structure of the agent's decision processes. The result is that complex chains and feedback loops can be constructed from relatively simple services using metadata.

To illustrate the power of Evaluators, in Figure 3 we show the evaluation chain that is used to produce sales quotas and set prices in the MinneTAC configuration that ran in the 2007 competition. Each of the cells in this diagram is an Evaluator. Across the top of the diagram is a set of evaluators that estimate current market prices, future price trends, and the shape of the customer order-probability function, based on the method of "economic regimes" developed by Ketter [22].

We have implemented three different economic regime identification and prediction methods, namely Markov prediction (MP), Markov correction-prediction (MCP), and an exponential smoother lookup (ExpS) process, with the help of evaluators. We also designed a training data evaluator, which is shared by the individual regime evaluators. The training data evaluator uses an external data source that contains an analysis of a large number of past simulations. The analysis was developed using machine learning methods, as described in [22]. These evaluators can dynamically select the most appropriate portions of the training data for a given market situation. In a real

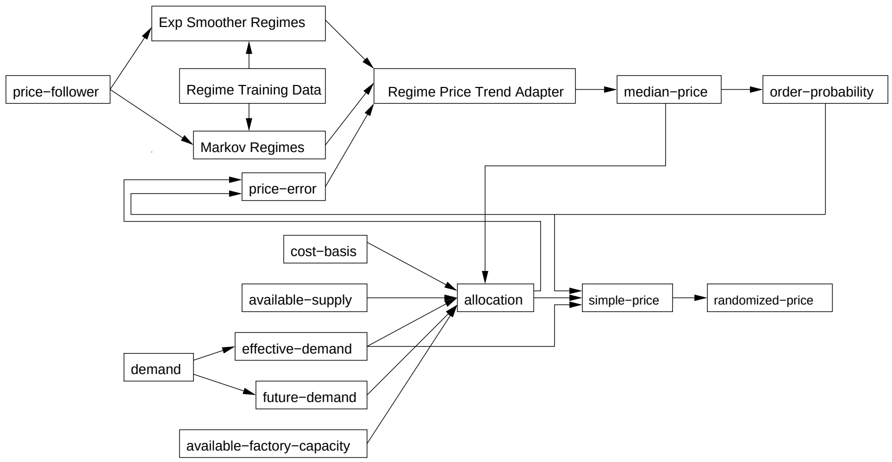

Fig. 3 Evaluator chain for a sales manager that uses sales quota and information provided by regimes to determine prices, price trends and order probability.

business network setting we would train the system on historical transaction data, and update it in regular intervals, e.g. after closing of a set of Dutch flower auctions.

The Sales component used with the evaluator chain shown in Figure 3 is conceptually simple – it places bids on each customer RFQ for which the randomized-price evaluator returns a non-zero value. The core of this chain is the allocation evaluator, which composes and solves a linear program each day of the simulation. The problem represents a combined product-mix and resource-allocation problem that computes daily sales quotas that maximize expected profit. The objective function is

$$\Phi = \sum_{d=0}^{h} \sum_{g \in \mathcal{G}} \Phi_{d,g} A_{d,g} \tag{1}$$

where Φ is the total profit over some time horizon h, $\mathcal{G}$ is the set of goods or products that can be produced by the agent, $\Phi_{d,g}$ is the (projected) profit for good g on day d, and $A_{d,g}$ is the allocation or "sales quota" for good g on day d. The constraints are given by the evaluators *available-factory-capacity*, the current day's *effective-demand*, projected *future-demand*, and by Repository data, such as existing and projected inventories of parts and finished products, and outstanding customer and supplier orders. Predicted profit per unit for each product type is the difference between *median-price* and *cost-basis* for those products.

Managers need to not only understand and control their decision processes, they also need to be able to visualize the data that are being used and produced by the elements of that process. This is very easy to do when decision processes are broken up into a set of discrete, single-purpose services. Figure 4 is a screen shot of an early prototype of the user interface.

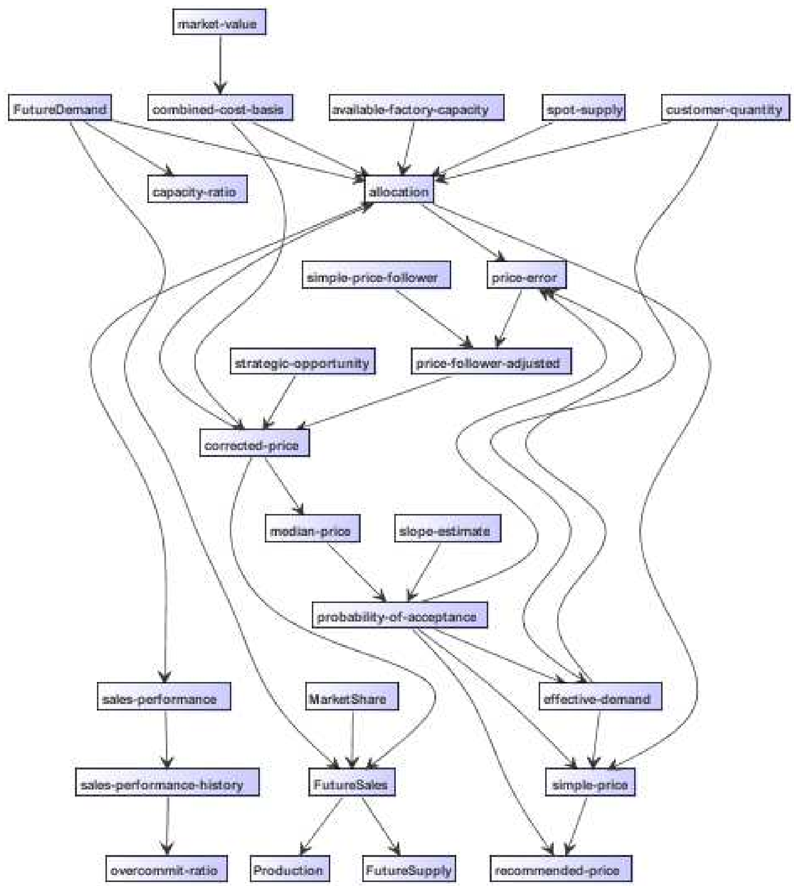

Fig. 4 Detail view from evaluator business network visualization tool.

Figure 5 displays the history of daily demand (the output of the "demand" evaluator) along with daily sales quotas (the output of the "allocation" evaluator). This information can be displayed for the overall market, or for individual products or market segments.

Figure 6 shows current sales commitments that have not yet been scheduled for production. The MinneTAC design allows a user to dynamically compose such "dashboard" displays by connecting a variety of graphing and plotting widgets to the outputs of the various evaluators. This can be done "on the fly", while the system is running, because the composition of services [35, 41] and visualizations is entirely dynamic.

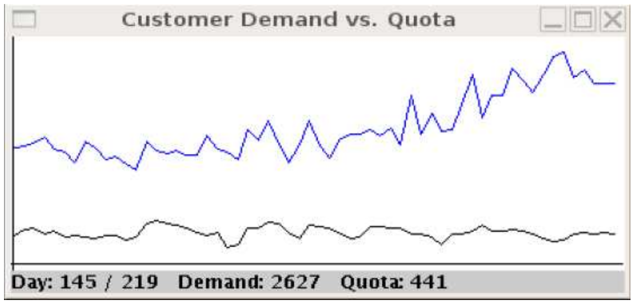

Fig. 5 Dynamic network status visualization: daily demand and sales quotas.

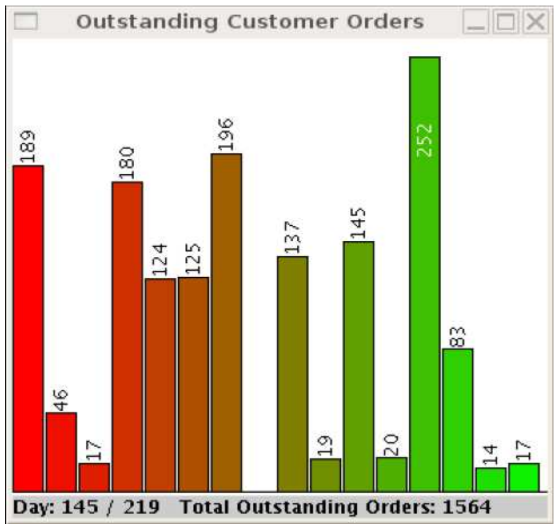

Fig. 6 Dynamic network status visualization: outstanding customer orders.

4 Related Literature

This work draws from several fields. In Computer Science, it is related to Software Engineering, Artificial Intelligence, autonomous agents, and multi-agent systems, especially architecture, machine learning, planning, and reasoning under uncertainty. In Economics and Information Decision Sciences, it draws from the framework of smart business networks, and decision theory. From Operations Research, it draws from work in combinatorial optimization and supply-chain management.

4.1 Multi-Agent Systems

Most agent design efforts have focused on either the autonomous behavior aspects of agency, or on interaction among agents. Norman *et al.* [29] describe *agent societies* that model organizational structures and automate business processes. These ADEPT agents negotiate over service agreements that can involve many parties and many dimensions. JADE [27] is an agent framework that has been used to build trading agents. Its primary emphasis is on building multi-agent systems that comply with FIPA specifications for inter-agent communications, and with flexible deployment in a network environment. These features are not necessary for the TAC SCM domain.

Vetsikas and Selman [38] describe a method for studying design tradeoffs in a trading agent. This approach could be used effectively in MinneTAC, but the issues addressed by their method are orthogonal to the component/evaluator scheme underlying MinneTAC. Vytelingam *et al.* [39] describe the IKB approach to the design of trading agents, consisting of an Information layer, a Knowledge layer, and a Behavioral layer. Podobnik *et al.* [31] have applied this approach to the TAC SCM scenario in CrocodileAgent. The MinneTAC design could be roughly mapped to this scheme, with the database as the Information layer, the set of evaluators as the Knowledge layer, and the decision components as the Behavioral layer.

He *et al.* [12] have adopted a design consisting of three internal "agents" to handle Sales, Procurement, and Production/Shipping. Sales decisions use a fuzzy logic module. Some algorithmic aspects are given, but there is little further detail on the architecture of the agent. TacTex05, the winner of the 2005 competition [30] is based on two major modules, a Supply Manager that handles procurement, and a Demand Manager that handles sales, production, and shipping. These modules are supported by a supplier model, a customer demand model, and a pricing model that estimates sales order probability.

4.2 Smart Business Networks

During the mid-nineties Goldman et al. [11] and Sanchez [32] stressed that in highly dynamic business networks the capability of a quick connect of network actors (businesses) is essential to enable fast response times and greater variety when presented with new product opportunities. The concept of "quick connect" includes a search and select behavior by the different businesses. Goldman et al.[11] further argue the need for a "quick disconnect" when the business transaction is over, otherwise open network connections can create unwanted information flows that make create unwanted side effects. At the time these articles were published no such network existed. Our architecture offers a unique way of automatically connecting, disconnecting and communicating with the appropriate actors in the network.

One has to pay extra attention to the interfaces of the different network actors. Establishing a temporary connection between actors needs to be grounded on a good and matching interface design. This interoperability can be facilitated by modularity. Garud et al.[10] define modularity as decomposability of a system by grouping elements into a smaller number of subsystems. Modularity is further a very well known concept in the software engineering field, and it refers to the extent to which software is divided into components, called modules, which have high internal cohesion[2], low coupling[3] between each other, and simple interfaces. Our architecture exhibits high cohesion and low coupling.

Hoogeweegen et al.[15] and van Liere [26] argue that knowledge of the network structure empowers the decision maker, and leads to better business decision. With our approach we are able to visualize the network structure, and even drill down on particular network actors to get a detailed picture of specific decision chains. Kambil and Short[19] already argued in 1994 that there is a strong need to construct software tools for business network representation, visualization, and analysis. These tools can help researchers and managers to visualize the different network actors, or roles, and linkage-based strategies of different organizations enabling the systematic representation and analysis of changes in emerging organizational forms. Our architecture offers unique capabilities for network visualization, role- and linkage analysis.

Creating performance and information dashboards [8] is part of the new emerging field of Business intelligence (BI) [34]. BI is a very powerful tool, as it provides functionalities such as real-time monitoring, performance reporting and support for exploring solution space with normative models, statistical techniques and visualization. Business intelligence software can crawl the web, mine data, and come back with a report customized to user preferences. Our architecture fully supports BI and our dashboards are customizable for individual managers. According to Adam and Pomerol [1] the layout of an economic dashboard has a direct impact on the understanding derived by managers. We believe that our customizable design will facilitate managerial decision making. They argue that a graphical user interface (GUI) of a dashboard should be leveraged to maximize the visual impact of the dashboard.

Furthermore dashboards (i) provide users with functions to find more detailed information of a certain metric or indicator (drill-down capabilities), (ii) provide users an interactive way of communicating with different actors (agents) of the network, (iii) allow customizing the appearance of how information is delivered by the agents and their granularity (days vs. weeks vs. months views), and (iv) provide search queries which helps agents to learn from a user. A complete and extensive work on the visual design of dashboards has been presented by Few [9]. According to Few many software com-

[2] A measure of the extent to which related aspects of a system are kept together in the same module, and unrelated aspects are kept out. High cohesion is better than low cohesion.

[3] A measure of the extent to which interdependencies exist between software modules. Low coupling is better than high coupling.

panies have developed and sold dashboard applications since 2001. This year was characterized by the Enron scandal that increased awareness throughout companies to closely monitor their most important business processes. Software companies from all kinds of sizes, such as Microsoft and Oracle, have developed dashboards[4].

5 Conclusions and Future Work

Experimental work with multi-agent systems in business networks requires an implementation. Often, the design qualities that best support experimental work are different from those normally considered "ideal" in industry. In complex economic scenarios such as TAC SCM, the desired design qualities include clean separation of infrastructure from decision processes, ease of implementation of multiple decision processes, clean separation of different decision processes from each other, and controllable generation of experimental data. In a competition environment, the ability to compose multiple agents with different combinations of decision process implementations makes it possible to test hypotheses about the effectiveness of competing decision models.

We have presented one way to construct such an agent, using a readily-available component framework[5] and a facility that allows metadata-driven composition of analysis and modeling tools using Evaluators. Additionally we presented tools to visualize the network structure, and economic dashboards to present the current state of each business unit.

There are many possible extensions to the basic design we presented here. One that we are currently pursuing is to add an "executive" component to allocate "resources" to competing implementations of basic decision processes within a single agent. This would allow a high degree of adaptability in the game environment, where the level of demand can fluctuate greatly, and where the actions of other agents can have a significant impact on the markets.

As implementation of business intelligence requires a lot of time, money and effort, managers need to know when to consider business intelligence and when not. We implemented our approach in TAC SCM, an abstraction of a real world supply-chain scenario to circumvent real world issues like economic risk, etc. The next step is to create a web service wrapper around the evaluators, and integrate it in a real business network, such as the Dutch Flower auction [17, 18].

We plan to implement automated web services [35, 41] to better connect to unknown network actors to guarantee a smooth run of the network as suggest by [14].

[4] `http://www.enterprise-dashboard.com`

[5] We used the Apache Excalibur component framework, see `http://excalibur.apache.org/`.

Acknowledgements Partial support for Maria Gini is gratefully acknowledged from the National Science Foundation under award NSF/IIS-0414466.

References

1. Adam, F., Pomerol, J.: Critical Factors in the Development of Executive Systems–Leveraging the Dashboard Approach. Decision Making Support Systems: Achievements and Challenges for the New Decade (2002)
2. Berry, P., Conley, K., Gervasio, M., Peintner, B., Uribe, T., Yorke-Smith, N.: Deploying a personalized time management agent. In: Proc. of the Fifth Int'l Conf. on Autonomous Agents and Multi-Agent Systems. Hakodate, Japan (2006)
3. Chopra, S., Meindl, P.: Supply Chain Management. Pearon Prentice Hall, New Jersey (2004)
4. Collins, J., Arunachalam, R., Sadeh, N., Ericsson, J., Finne, N., Janson, S.: The Supply Chain Management Game for the 2006 Trading Agent Competition. Tech. Rep. CMU-ISRI-05-132, Carnegie Mellon University, Pittsburgh, PA (2005)
5. Collins, J., Bilot, C., Gini, M., Mobasher, B.: Decision processes in agent-based automated contracting. IEEE Internet Computing **5**(2), 61–72 (2001)
6. Collins, J., Ketter, W., Gini, M.: Architectures for Agents in TAC SCM. In: AAAI Spring Symposium on Architectures for Intelligent Theory-Based Agents, pp. 7–12. Stanford University, Palo Alto, California (2008)
7. CombineNet: Sourcing solutions . http://www.combinenet.com/sourcing_solutions/ (2006)
8. Eckerson, W.W.: Performance Dashboards: Measuring, Monitoring, and Managing Your Business. Wiley (2005)
9. Few, S.: Information Dashboard Design: The Effective Visual Communication of Data. O'Reilly Media, Inc. (2006)
10. Garud, R., Kumaraswamy, A., Langlois, R.: Managing in the Modular Age: Architectures, Networks, and Organizations. Blackwell Publishers (2002)
11. Goldman, S., Nagel, R., Preiss, K.: Agile competitors and virtual organizations. Van Nostrand Reinhold New York (1995)
12. 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, pp. 1159–1166 (2006)
13. van Heck, E., Vervest, P.: Smart business networks: how the network wins. Commun. ACM **50**(6), 28–37 (2007). DOI http://doi.acm.org/10.1145/1247001.1247002
14. van Hillegersberg, J., Boeke, R., van den Heuvel, W.: Potential of Webservices to enable smart business networks. Journal of Information Technology **19**(4), 281–287 (2004)
15. Hoogeweegen, M., van Liere, D., Vervest, P., van der Meijden, L., de Lepper, I.: Strategizing for mass customization by playing the business networking game. Decision Support Systems **42**(3), 1402–1412 (2006)
16. I2: Next-generation planning. http://i2.com/solution_library/ng_planning.cfm (2006)
17. Kambil, A., van Heck, E.: Reengineering the Dutch Flower Auctions: A Framework for Analyzing Exchange Organizations. Information Systems Research **9**(1), 1–19 (1998)
18. Kambil, A., van Heck, E.: Making Markets: How Firms Can Design and Profit from Online Auctions and Exchanges. Harvard Business School Press (2002)
19. Kambil, A., Short, J.: Electronic integration and business network redesign: a roles-linkage perspective. Journal of Management Information Systems **10**(4), 59–83 (1994)
20. Kearns, M., Ortiz, L.: The Penn-Lehman Automated Trading Project. IEEE Intelligent Systems pp. 22–31 (2003)

21. Keller, P.W., Duguay, F.O., Precup, D.: Redagent - winner of the TAC SCM 2003. SIGecom Exchanges **4**(3), 1–8 (2004)
22. Ketter, W.: Identification and prediction of economic regimes to guide decision making in multi-agent marketplaces. Ph.D. thesis, University of Minnesota, Twin-Cities, USA (2007)
23. Ketter, W., Collins, J., Gini, M., Gupta, A., Schrater, P.: A predictive empirical model for pricing and resource allocation decisions. In: Proc. of 9th Int'l Conf. on Electronic Commerce, pp. 449–458. Minneapolis, Minnesota, USA (2007)
24. Ketter, W., Collins, J., Gini, M., Gupta, A., Schrater, P.: Detecting and Forecasting Economic Regimes in Multi-Agent Automated Exchanges. Decision Support Systems **in publication** (2008)
25. 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, pp. 208–217. Ann Arbor, USA (2006)
26. Liere, D.: Network Horizon and the Dynamics of Network Positions: A Multi-Method Multi-Level Longitudinal Study of Interfirm Networks (2007)
27. Moraitis, P., Petraki, E., Spanoudakis, N.: Engineering JADE agents with the Gaia methodology. In: R. Kowalszyk, J. Miller, H. Tianfield, R. Unland (eds.) Agent Technologies, Infrastructures, Tools, and Applications for e-Services, *Lecture Notes in Computer Science*, vol. 2592, pp. 77–91. Springer-Verlag (2003)
28. Niu, J., Cai, K., Parsons, S., Gerding, E., McBurney, P., Moyaux, T., Phelps, S., Shield, D.: JCAT: A Platform for the TAC Market Design Competition. In: Proceedings of the Seventh Int'l Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS 2008), pp. –. Estoril, Portugal (2008)
29. Norman, T.J., Jennings, N.R., Faratin, P., Mamdani, E.H.: Designing and implementing a multi-agent architecture for business process management. In: M.J. Wooldridge, J.P. Müller, N.R. Jennings (eds.) Intelligent Agents III, *Lecture Notes in Artificial Intelligence*, vol. 1193, pp. 261–275. Springer-Verlag, Berlin (1997)
30. Pardoe, D., Stone, P.: Tactex-05: A champion supply chain management agent. In: Proc. of the Twenty-First National Conference on Artificial Intelligence, pp. 1389–1394. AAAI, Boston, Mass. (2006)
31. Podobnik, V., Petric, A., Jezic, G.: The crocodileagent: Research for efficient agent-based cross-enterprise processes. In: R. Meersman, Z. Tari, P. Herrero (eds.) On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, vol. 4277, pp. 752–762. Springer-Verlag (2006)
32. Sanchez, R.: Strategic Flexibility in Product Competition. Strategic Management Journal **16**, 135–159 (1995)
33. Sandholm, T.: Expressive commerce and its application to sourcing: How we conducted $35 billion of generalized combinatorial auctions. AI Magazine **28**(3), 45–58 (2007)
34. Shmueli, G., Patel, N., Bruce, P.: Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner. Wiley-Interscience (2006)
35. Sirin, E., Hendler, J., Parsia, B.: Semi-automatic composition of web services using semantic descriptions. Web Services: Modeling, Architecture and Infrastructure" workshop in conjunction with ICEIS2003 (2002)
36. Vervest, P., van Heck, E., Preiss, K., Pau, L.F.: Smart Business Networks. Springer Verlag, Berlin (2005)
37. Vervest, P., Preiss, K., Heck, E., Pau, L.: The emergence of smart business networks. Journal of Information Technology **19**(4), 228–233 (2004)
38. Vetsikas, I.A., Selman, B.: A principled study of the design tradeoffs for autonomous trading agents. In: Proc. of the Second Int'l Conf. on Autonomous Agents and Multi-Agent Systems (2003)
39. Vytelingam, P., Dash, R.K., He, M., Jennings, N.R.: Trading strategies for markets: A design framework and its application. In: H.L. Poutré, N.M. Sadeh, S. Janson (eds.)

Agent-Mediated Electronic Commerce: Designing Trading Agents and Mechanisms, *Lecture Noted in Artificial Intelligence*, vol. 3937, pp. 171–186. Springer-Verlag (2006)

40. Wellman, M.P., Wurman, P.R., O'Malley, K., Bangera, R., Lin, S., Reeves, D., Walsh, W.E.: Designing the market game for a trading agent competition. IEEE Internet Computing **5**(2), 43–51 (2001)
41. Wu, D., Parsia, B., Sirin, E., Hendler, J., Nau, D.: Automating DAML-S web services composition using SHOP2. Proceedings of 2nd International Semantic Web Conference (ISWC2003) (2003)