Using a Trust Model in Decision Making for Supply Chain Management

Yasaman Haghpanah and Marie desJardins
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore MD 21250
{yasamanhj, mariedj}@umbc.edu

Abstract

One of the critical factors for a successful cooperative relationship in a supply chain partnership is trust. Many real-world applications, such as Supply Chain Management (SCM), can be modeled using multi-agent systems. One shortcoming of current SCM models is that their trust models are ad hoc and do not have a strong theoretical basis. As a result, they are unable to model subtleties in agent behavior that can be used to build a more accurate trust model. We propose a trust model for SCM that is grounded in probabilistic game theory. In this model, trust can be gained through direct interactions and/or by asking for information from other trustworthy agents. We will use this model to simulate and study supply chain market behavior.

Keywords

supply chain management, multi-agent systems, decision theory, game theory

Introduction

Almost all organizations need trust in order for the agents within them to become more successful in their relationships with their partners. In supply chain management, establishing trust improves the chances of a successful supply chain relationship, and increases the overall benefit. One of the primary purposes of supply chain management applications is to assist an organization to respond to events in a synchronized and timely fashion. Application domains include e-commerce and e-business applications.

Supply chain networks have often been modeled in the research literature with multi-agent systems in which the agents need to cooperate with one or more partners. This collaboration becomes more effective when agents have the ability to choose their partners based on the trustworthiness of the candidates. Trust is defined as the belief an agent has that the other party will fulfill its promises, given the possibility that the partner may defect to get higher benefits (Dasgupta 1998). A major shortcoming in previous research on trust in supply chain management is that the trust-based decision making is not grounded in a formal trust model.

In this work, we propose a trust model for SCMs that incorporates trust factors specific to SCM, represented in probabilistic and utility-based terms. To do this, we adapt the HAPTIC model, presented by Smith and desJardins (2009), which is grounded in game theory and probabilistic modeling. This model has been applied to two-player Iterated Prisoner's Dilemma (IPD) games. Using decision and game theory, our model builds cooperative agents for supply chain management applications with uncertainties and dynamics. Our ultimate goal is to have a complete and sound trust model for SCM using game and decision theory by combining the HAPTIC model with SCM-related trust factors and a reputation exchange mechanism in order to adapt it to real-world scenarios. Our model takes into account the effects of variations over interactions such as variable payoffs. We will also investigate how market behavior is affected by different trust factors. We will identify optimal strategies for different situations; i.e., those strategies that result in the best performance and overall returns. In this paper, we present our initial framework for SCM and the incorporation of the theoretic model, which is currently under development.

The HAPTIC Model

The HAPTIC model is a trust-based decision framework that allows an agent to predict a partner's actions in the current trade and use these predictions to decide whether or not to trust that partner. The key insight in HAPTIC is that it separately models *competence* and *integrity*. Competence is modeled as the probability that a given agent will able to execute an action in a particular situation. Integrity is an agent's attitude towards honoring its commitments, and is affected by the perceived probability of future interactions. In the HAPTIC framework, an agent observes the behavior of agents and estimates their competence and integrity. It then uses this learned information for decision making in future interactions with the same agent. It is important to note that the HAPTIC framework distinguishes between competence failures and integrity failures. When an agent defects in its action and a failure occurs, it is important for the other agent involved in this interaction to understand whether this failure was due to incompetence of an honest agent, or the result of cheating of a competent agent with low integrity. By using variable payoffs, this ambiguity can be resolved. An honest but incompetent agent should defect randomly, irrespective of the payoff. By contrast, a cheating agent will show a pattern in its defects; this pattern will be directly related to the payoff of the interactions.

Agents' interactions are modeled using the Harsanyi transformation from game theory, which converts a game with incomplete information to a strategic game in which players may have different types and are uncertain about their opponent's type. In this Bayesian game, the uncertainty is described by a probability distribution over all possible player types. Learning for each agent occurs by updating this probability distribution based on the observed play after each interaction with their partner. A learning HAPTIC player interprets the outcome of plays after each round as either a Success or a Failure, based on its hypothesis about that agent's type. Iterative games between two agents allow HAPTIC players to quickly reduce the set of probable types being considered. The HAPTIC learning method uses observations of agent behavior to estimate the trust factors of each agent. This learning process in this model is faster than many of the known agent frameworks (Smith and desJardins 2009).

We adapt the HAPTIC learning process for our SCM model. From the learning perspective, our model differs with HAPTIC because we use utility instead of success or failure as the outcome of each interaction.

We next present a brief literature review of trust frameworks and models in SCM, then explain our approach to trust and reputation in SCM. Finally, our future work and conclusions are outlined.

Related Work

Trust has been used in all levels of multi-agent interactions, including individual-level and system-level trust. In individual-level trust, each agent has some beliefs about the honesty and reliability of its counterparts, which can be formed through direct interaction with partners, by asking other agents about potential candidates (reputation exchange), and/or by forming and reasoning about beliefs of other agents' characteristics. System-level trust can be achieved when the rules that control the system force agents to be trustworthy (Ramchurn et al. 2004); however, this approach is not practical for most real-world applications. Learning trust from direct experience is advantageous when agents have repeated interactions. Conversely, reputation exchange is most useful for learning the trustworthiness of other agents quickly (Mui 2003).

There have been several proposed approaches for adding trust models into SCM. Centeno et al. (2009) propose a reputation mechanism based on organizational concepts and personal norms, with which agents define their preferences about potential interactions. However, this information is not sufficient for adaptively learning trust models, since agents do not model their confidence in the information they receive from other agents. Lin et al. (2005) build a trust model based on experiences with suppliers; trust is measured in terms of product quality, order-cycle time, and price. They generalize these factors to the abstract concepts of ability, integrity, and benevolence. This model does not use probabilistic decision theory. Other SCM trust factors

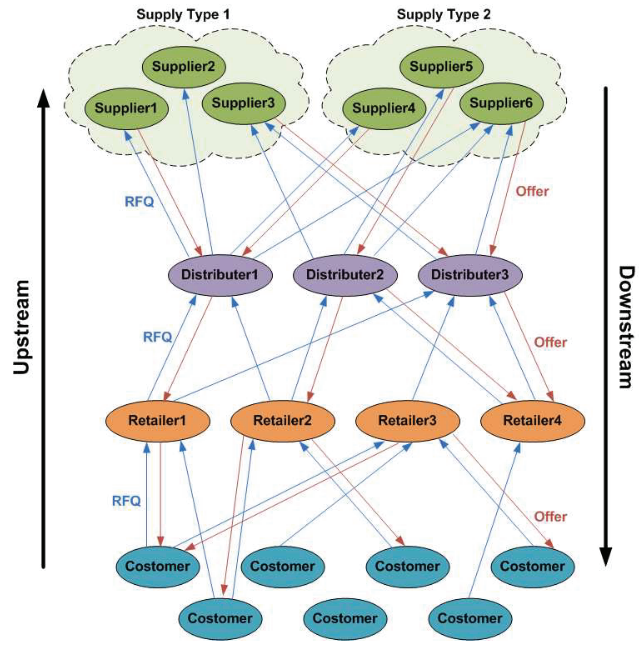

Figure 1: Example of SCM

have been studied as well, although many of them are focused on specific SCM industries. For example, Paterson et al. (2008) studied twelve trust factors, identifying three factors that are critical to the horticulture supply chain: shared values, point-of-sale information, and honesty and integrity.

Buffett and Scott (2004) propose a model using the request-for-quote (RFQ). In their model, the buyer specifies the quantity needed and the desired delivery date. The authors consider only indirect costs, such as the time taken to compute optimal RFQs, and the possibility of being neglected by suppliers when failing to respond to their requests repeatedly. They do not consider any direct costs associated with requesting RFQs. In this model, the suppliers use a simplified reputation model to decide which RFQs to process first. This reputation model uses the ratio of the quantity requested by an agent to the quantity actually ordered, over the entire game. Our model differs from Buffett and Scott's in that they do not have a trust model and their reputation model is simple and ad hoc. In addition, we consider direct costs as well as indirect costs.

The above trust frameworks categorize some aspects of trust, but most of them focus on producing a single metric for trust or reputation (Sabater and Sierra 2001, Mui 2003). In addition, many trust frameworks do not use probabilistic methods, preferring ad hoc valuation schemes (Sabater and Sierra 2001).

In this paper, we explain how we combine this model with SCM-related trust factors and a reputation exchange mechanism to adapt it to real-world scenarios.

Approach

Our SCM model consists of several layers in a supply network, where each layer contains a number of agents. The

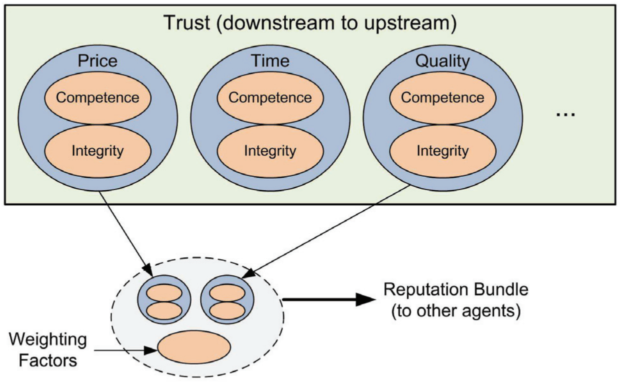

Figure 2: Trust components and an example of a reputation bundle

layers can correspond to suppliers, producers, distributors, or retailers. Each agent in each level connects to some of the agents in neighboring levels to obtain or provide services, ultimately forming a team or "supply chain." In general, upstream agents provide services (or offers) to adjacent downstream agents, and downstream agents ask for services or send RFQs to the adjacent upstream agents, as shown in Figure 1. In this model, we use variable payoffs for different services in different environments. Agents in this framework use a utility function to estimate the future reward that would result from working with a potential partner. This utility function is calculated based on the amount of benefit minus the cost of the transaction.

We consider personal criteria or preferences in the team formation process of SCM. Each downstream agent has a list of criteria and preferences for the services or goods that it needs. For example, one downstream agent might need a high-quality material from an upstream agent, with three weighted criteria: quality 70%, price 20%, and time 10%. In this case, the most important factor for the agent is quality. Downstream agents send a RFQ to upstream agents. The downstream agents will select the closest match of possible offers based on their criteria and preferences in such a way that the selected offer maximizes the agent's return utility.

In our model, trust by downstream agents in upstream agents is maximized when the latter agents provide goods and services with low prices and good quality in a timely manner. To model trust in this case, we define the two components of competence and integrity for each factor (e.g., quality, price, and time) as shown in Figure 2. The competence for each of these factors is the probability that the upstream agent is able to fulfill the commitment. Integrity is modeled as the degree to which the agent keeps the same behavior in the long term and in variable-payoff situations. For example, the upstream agent might offer the desired service for two rounds, but after gaining the trust of the downstream agent, the agent might betray in the third round, if they have low integrity for that service. Similarly, the trust of an upstream agent to a downstream agent is affected by the number of times that the downstream agent has accepted the upstream agents offer, the payoff level for each interaction, and the frequency of on-time payments. Each of these factors is also modeled using competence and integrity. The combination of these factors will yield an overall trust level of an upstream agent to a downstream agent. An upstream agent can give different offers (on the same trade element) to different downstream agents, since it might have different levels of trust to them based on their competence and integrity. Also, it might accept an RFQ from one downstream agent and not accept the same RFQ from another downstream agent (due to a higher level of trustworthiness in the first agent).

We propose to add another individual-level trust mechanism—namely, reputation exchange—into our model. Agents might have different opinions about the reputation provided by another agent, based on how well each agent knows that agent. To address this complexity, we propose to use a weighting factor for the exchanged information, taking into account the size of payoffs in the previous transactions with that agent, how many interactions they have had, and for how long they have known each other.

Reputation can be exchanged as a bundle including trust factors (quality, price, time, and/or on-time payment), confidence in trust factors (competence and integrity), and personal criteria, as shown in Figure 2. To make our model more realistic, we will consider cost associated with reputation exchange, modeled as money, time, and/or a limitation on the number of messages each agent can send.

Individual components of a reputation bundle can be shared with other agents, based on the need for certain information (i.e., partial reputation exchange when agents only ask for partial information of other agents). Consider a team that is almost ready to complete its task, but suddenly one of members fails due to inclement weather, disasters, or even betrayal. In this case, the team should look for an agent who is offering a high level of competence for time in its service. At that point the team may not have enough time (because of time constraints) to spend on finding a perfect partner (having both high competence and high integrity for all of the trust factors), so it will only ask other agents for partial information (which is a high competence in time in this case). In other scenarios, an agent might need just the high integrity in quality, so it asks for agents who offer quality with high integrity.

In our proposed model, agents initially use reputation exchange more than their personal experiences, since they have not yet had interactions with many agents. After a few rounds of interactions, agents get a sense of whom they can trust in the long run, and can start updating their trust model based on their personal experiences. As a result, the use of reputation exchange decreases, as shown in Figure 3.

Selecting partners from one layer to another can happen in both directions: downstream to upstream agents, and vice versa. In our model, we assume that partner selection happens from downstream to upstream agents, since we found this model closer to the real world (i.e., a customer selects its distributer, retailer or supplier).

We use an RFQ model, similar to the work of Buffett and Scott. In our model, the downstream agent provides a preference list to the upstream agent. There are several costs as-

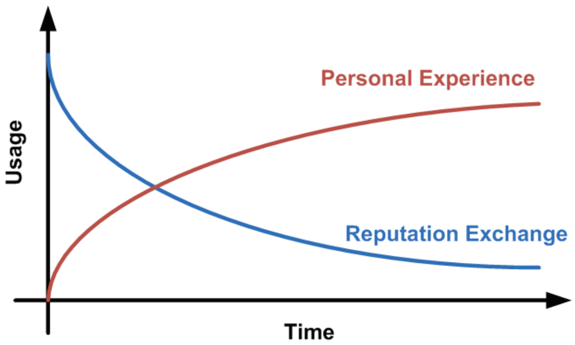

Figure 3: Level of usage of personal experience and reputation exchange in trust calculation over time

sociated with RFQs: the time taken to send request and the effects on trust and reputation when the downstream agent (RFQ sender) repeatedly reject offers from upstream agents, as a result of too many sent RFQs. We also consider a utility cost associated with the way in which agents send their RFQs. For example, sending RFQs by email is free but posting them by mail is not.

Initialization

- The SCM life cycle starts when an initial set of requests and preferences are generated and assigned to different customers.
- The unit of time for our model is one day, and we run the simulation for one year.
- Different services with associated prices are assigned to the suppliers.
- A random number of requests and preferences are generated and assigned to random customers.
- A partial view of the upstream services is available to the downstream agents, to make our model more like a real-world scenario, in which a customer has only limited knowledge of the available providers.
- Agents in each layer will be assigned an equal amount of credit to use.

Flow of events

The customers send RFQs, along with their preferences, to some of their visible upstream agents, who offer those services. Those upstream agents then send RFQs to their visible upstream agents, until they reach the suppliers. Upstream agents calculate the profit for each requested service based on the trust it has in the requester, and prepare responses for the downstream agents. Each agent chooses the best offer from its upstream agents, based on its preferences. The agent then calculates its own profit, adds to the cost of service and generates a new offer for its downstream agent.

Once an offer is accepted by a downstream agent, the agents are committed to complete that order. A penalty, which is defined in each RFQ, will be assigned for each side in case of a cancellation or date change.

Agent's responsibilities

In this model, we define three roles for each agent: customer, upstream, and downstream agents. Each agent can act simultaneously in the roles of both downstream and upstream agent. The customers are the agents who does not have downstream agents. They have the responsibilities of a downstream agent as well as a customer's responsibilities. Customers are responsible to:

- Submit requests, along with their preferences, to upstream agents.

Downstream agents are responsible to:

- Select partners from upstream agents.
- Send RFQs to upstream agents.
- Receive offers from upstream agents.
- Select the best offer received.Downstream agents consider the offers from upstream agents only if the service is consistent with the downstream agent's preferences (within its budget range and not overdue).

Upstream agents are responsible to:

- Receive RFQs from downstream agents.
- Process RFQs (accept or reject them).
- Calculate the profit for each offer.
- Generate offers in response to downstream agents' RFQs.

Besides the above responsibilities, each agent has these duties:

- Calculate trust, and update the agent's competence and integrity for each of the trust factors after each interaction (acceptance or rejection of a RFQ).
- Calculate the profit in each transaction as well as calculating the overall cumulative profit.

Utility Calculation

In our proposed SCM model, each agent aims to maximize its utility. The utility of each agent is equal to the profit they make in each interaction, calculated as:

$$P_T = O_T - C_T - C_{RFQ}, \tag{1}$$

where in each transaction T, P_T is profit, O_T is the offering price, C_T is the cost of the service, and C_{RFQ} is the cost of all RFQs sent for this service.

$$P_{total} = \sum_T P_T, \tag{2}$$

where P_{total} is the total profit of all transactions.

The amount of added profit in each offer depends on the trust level: each agent can choose the amount of profit to add to the offering price.

The utility is defined in the upstream-to-downstream flow as in Equations (1) and (2). The only exception is the customer utility calculation, since there are no downstream agents. In this case, we define a utility based on the customer's satisfaction with the deal, and how close the deal is to its preferences.

Decision Making

Decision making happens when an agent wants to send and process RFQs, generate offers, and select the best offer. In general, agents make decisions by computing expected returns and choosing the action with the highest expected value, using their knowledge of other agents' competence and integrity, the current round's payoffs, and the expected average stakes of future rounds.
Decision making for downstream agents:

- To send RFQs, each downstream agent calculates the profit and return of potential services offered by upstream agents. Also, it takes into account the level of trust it has in that agent (taking into account how many transactions they had, what were the payoffs, and what is its current hypothesis of the competence and integrity of that agent).
- To select the best offer, a downstream agent selects the closest match of available offers based on their criteria and preferences in such a way that the selected offer maximizes the agent's return utility.

Decision making for upstream agents:

- To generate an offer, an upstream agent calculates the return and profit for each service and takes into account to which agent the offer goes. To do this, it needs to consider how much trust it has in the downstream agent and how many times the downstream agent has rejected or accepted its offers.
- To process RFQs, an upstream agent calculates the return and profit from the RFQs. They give preference to those downstream agents who have a better record of accepting the upstream agent's offers.

Conclusions and Future Work

In this paper, we presented a proposed trust model to be incorporated into a realistic SCM agent-based model. We intend to extend the HAPTIC decision-theoretic trust model into our SCM simulation model, by defining SCM-specific trust factors (e.g., quality, time, and price), and by incorporating indirect (reputation-based) trust. This proposed work is currently under development. We claim that our model will help to increase (or maximize) the overall profit of the supply chain over time.

We will investigate how different trust factors affect the system in terms of performance and stability in realistic markets under different conditions. We also plan to consider a feedback mechanism from downstream to upstream agents on the given services. This will allow us to simulate the real-world review process widely used in different markets.

The learning process we use in this model is fast and since the agents are engaged in information sharing (reputation exchange), we expect the market to reach stability and equilibrium quickly. If an adverse event happens (such as inclement weather, a system crash, or betrayal), then the system should recover and reach a new stability in a timely manner. Therefore, we expect the system to work well under uncertain and dynamic conditions.

References

Buffett, S., and Scott, N., 2004. "An Algorithm for Procurement in Supply Chain Management," *AAMAS Workshop on Trading Agent Design and Analysis,* 9-14.

Centeno, R., Silva, V., and Hermoso, R., 2009. "A reputation model for organisational supply chain formation," *COIN@AAMAS*.

Dasgupta, P. 1998. "Trust as a commodity," *Trust: Making and Breaking Cooperative Relations,* chapter 4, 49-72.

Lin, F., Sung, Y., and Lo, Y. 2005. "Effects of trust mechanisms on supply-chain performance: A multi-agent simulation study," *Int. J. Electron. Commerce* 9(4):9-112.

Mui, L. 2003. "Computational models of trust and reputation: Agents, evolutionary games, and social networks," *PhD thesis,* Massachusetts Institute of Technology.

Paterson, I., Maguire, H., and Al-Hakim, L. 2008. "Analysing trust as a means of improving the effectiveness of the virtual supply chain," *Int. J. Netw. Virtual Organ.* 5(3/4):325-348.

Ramchurn, S. D., Huynh, T. D., and Jennings, N. R. 2004. "Trust in multi-agent systems," *The Knowledge Engineering Review* 19(1):1-25.

Sabater, J., and Sierra, C. 2001. "Social regret, a reputation model based on social relations," In *ACM SIGecom Exchanges* 3.1:44-56.

Smith, M. and desJardins, M. 2009. "Learning to trust in the competence and commitment of agents," *Autonomous Agents and Multi-Agent Systems* 18(1):36-82.