# Using Agent Similarities in Business Rules for the Supply Chain

Ioan Alfred Letia
Technical University of Cluj-Napoca
Str. Baritiu nr. 28, RO-400391, Cluj-Napoca
Romania
*letia@cs.utcluj.ro*

Adrian Groza
Technical University of Cluj-Napoca
Str. Baritiu nr. 28, RO-400391, Cluj-Napoca
Romania
*adrian@cs-gw.utcluj.ro*

*Abstract* – **The continuing globalization imposes new challenges to the "e-supply chain", among them the need of agents to react properly in case of cooperation or otherwise, according to environment changes. Our framework aims to find agents with similar business rules, with environment perception and its influence represented by causal relations. Then we try to find the unique strategy for these appropriates agents. The causal links are translated in RuleML derivation rules, to be used in decision making regarding other agents for long term business relationship. The framework is aimed to distinguish between public knowledge regarding the business activity available on the Semantic Web, and private knowledge, known to company insiders.**

## I. INTRODUCTION

A supply chain can be defined as a network of autonomous or semi autonomous business entities collectively responsible for procurement, manufacturing and distribution activities [9]. All the business entities within a supply chain interact during the e-commerce life-cycle [11] through business rules [5]. To function properly, the members within a supply chain need some coordination when the market changes and the agents' decisions are dependent on (dis)similar business rules.

A goal in our study is to identify medium perceptions and reaction of individual agents. By identifying these perceptions, the agents are capable of uncovering how their business partners understand their world. First, the agent must determine that the ideas in two or more individual business strategies are similar enough to be represented by one idea in a supply chain relationship.

Consider the next two examples. On the one hand, a firm has to decide between two potential suppliers which provide quite similar offers for the moment. The firm's dilemma is with which partner the business relationship is more probable to be stable in time. This decision is significant, because any future change in the supply chain implies costs. We consider that a relevant factor for durability of the relationship is that some firms must react in a similar way to the environment changes.

On the other hand, suppose the supply chain is formed by agents with different strategies. For instance, the shop point wishes to increase number of clients by practicing low costs for its products. Meanwhile, the supplier wants to penetrate the market through high quality products, which imply high costs. This is opposite to the sell point market strategy. The losses affect both parties: the supplier might not sell its high quality items in shops with a different category of clients, while the shop will loose a percentage of its clients, who try to find a cheaper sell point.

While similarity criteria have been used to support decision making in negotiations, our interest here is beyond automated negotiation aiming at the life-cycle of the business processes. Although preferences have already been applied to negotiation, our intention is to capture strategic aspects regarding the coordination of actors in the business activity.

We present a method that increases the probability for the business relationship to continue in the future. We are doing this by observing how similar the agents react when the environment is changing. We provide also a method to find the common strategy of a supply chain formed by agents with different strategies.

## II. RELATED WORKS

RuleML [7] is a very promising technique for defining business rules, used for example in the project RACSA (Rule Applying Comparison Shopping Agent). It shows an e-commerce application enabled by the Semantic Web technology. The shopping web pages with the Semantic Web languages are marked by RACSA [8]. Based on their list prices, the real end prices are calculated by applying specific business rules, like calculation of gross/net prices, warranty time, transport costs, customer discounts (they appear in our framework as *reaction rules*).

Causal map representations of the relationships between agents' beliefs [2], based on a relational algebra, can be used for reasoning in the context of multiagent systems. They help to investigate aspects such as: reasoning on the subjective view in multiagent systems, qualitative distributed decision making and organization of agents considered as a holistic approach. We study a similar problem in the context of multiagent systems, trying to capture causality in RuleML and to provide a flexible framework for decision making in order to increase agents' coordination. The framework presented here is an attempt to fill the gap between simple reaction rules and how causality determines such rules.

## III. REPRESENTING BUSINESS RULES

In order the agents to communicate they have to share a common set of protocols and knowledge representation. In this section we describe how the agent internal state can be represented using RuleML. Every entity within the supply chain must act coherently according to the market strategy of the entire chain. In our scenario we distinguish the following types of possible strategies: 1) *Domination through costs*: the agents practice low costs in order to win; 2) *Domination through quality*: the agents offer a high quality to every client, including more services or more sell points; 3) *Niche strategy*: the agents focus on the specific target of clients.

For the first strategy, the agents try to maintain low costs of their products no matter how the environment changes. For instance, if demand decreases they prefer to react by decreasing the price and not by increasing the quality of their products. The agents who use the second strategy will prefer to increase the quality to handle demand fluctuations. Agents who use the last strategy will treat some of their clients preferentially by providing them discounts in order to maintain a constant rate of demand. Such different behaviors could cause conflicts in a long time business relationship.

### *A. Business rules in RuleML*

RuleML is a standard initiative that has its roots in the work of the Semantic Web, supported by both academia and industry. It is based on the XML syntax and provides a method for knowledge representation and reasoning.

The RuleML languages allow the exchange of the rules between distributed software components on the web or within large business corporations. The modular RuleML design contains a hierarchy of rules from reaction rules (event-condition-action rules), via integrity constraint rules (consistency-maintenance rules) and derivation rules (implication inference rules), to facts (derivation rules with no premise) [1]. Hence, we can specify queries and inferences in web ontologies, mappings between these ontologies or even dynamic web behaviors of workflows, services and agents.

*Reaction rules* have strong similarities with the behavior of active databases and they are based on the conceptual model of ECA (Event Condition Action). They represent the most important type of business rules according to [10]. They also allow to specify the agent behavior in response to browser events. *Integrity constraints* are considered as special reaction rules whose only possible kind of action is to signal inconsistency when certain conditions are fulfilled. *Derivation rules* are special reaction rules whose action happens to just add or assert a conclusion when certain premises are fulfilled. This assertion of conclusions can be regarded as a declarative step, as used for model generation and .x point semantics. Such rules can thus also be applied backward for proving a conclusion from premises. These rules are used to enhance the content of web pages or XML documents through dynamic inclusion of derived facts.

### *B. Agent internal state*

One advantage of using RuleML is that we can separate the business logic from the business object (as in fig. 1). Using this approach [3], [4], [6], a software agent may be specified by:

1) a RDFS-based taxonomy for defining the schema of its mental state; 2) a set of RDF facts for specifying its factual knowledge; 3) a set of RuleML integrity constraints for excluding nonadmissible mental states; 4) a set of RuleML derivation rules for specifying its terminological and heuristic knowledge; 5) a set of RuleML reaction rules for specifying its behavior in response to communication and environment events. The *reaction rules* are public; they form a facet of the agent which usually represents the basis for signing contracts or compare different offers on the web.

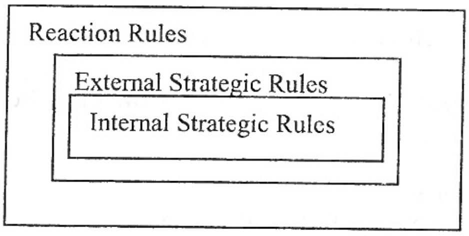


Fig. 1. Agent Architecture

When the context is to build a supply chain, these rules are not enough, as they do not guarantee at all a lasting relationship. Thus, an agent who cares about this aspect has to analyze deeper the behavior of its potential partners, by investigating the *external strategic rules* of the others agents. If these rules are not public they must be derived from the history. This is past events about the potential partner, included in the model through RuleML *facts*. Agent's *internal strategic rules* are reaction rules known in most cases only to the agent. They show how the agent performs its strategic reasoning and acquires beliefs. Using them together with a set of *integrity constraints*, the agent can derive its external strategic rules. *Reaction rules* are made public by each company in the style shown below.

**Reaction rules**
if client(gold) then client discount(0.5)
if client(silver) then client discount(0.4)
if payment(cash) then discount(0.3)

**External strategic rules**
if demand(+) then sellPrice(-)
if politicalStability(+) then quality(+)
if demand(0) then discounts(+)

They are used for a first selection, but they do not provide a real image of the potential partner. The reaction rules are derived directly from the external strategic rules. *External strategic rules* are used in our framework to estimate the strategy of potential actions, e.g., with "+" meaning *increase*, "0" meaning *remain constant* and "–" *decrease*. These rules are a direct consequence of how the agent builds its strategy using internal strategic reasoning rules. The representation of the first rule above in the syntax of RuleML is shown below.

**if demand(+) then sellPrice(-)**

```
<imp>
  < head> <atom>
    < opr><rel>sellPrice</rel></opr>
    <ind>decrease</ind>
  </atom> </ head>
  < body> <atom>
    < opr><rel>demand</rel></opr>
    <ind>increase</ind>
  </atom> </ body>
</imp>
```

*Internal strategic rules* form the internal state of the agent, which might reflect opposite types of reasoning.

**Internal strategic reasoning rules for two opposite agents** $S1$, $S2$

RULES for agent $S1$
if clientsNumber(+) then producedItems(+)
if producedItems(+) then unitProductionCost(-)
if unitProductionCost(-) then sellPrice(-)

RULES for agent $S2$
if clientsNumber(+) then demand(+)
if demand(+) then sellPrice(+)

At similar environment changes (*clientNumber( increase)*), the agent $S1$ makes the decision *sellPrice(decrease)*, while the agent $S2$ decides on the opposite one *sellPrice(increase)*. Such situations could influence the entire supply chain. Suppose most entities within the supply chain decide to decrease the prices, and some of them decide to increase it. The entire chain will be affected, and no agent might be able to finalize its market strategy. That is why it is important that supply chain members coordinate their market strategy. Having such a representation, if they decide to, the agents can share their business rules more easily.

## IV. FINDING AGENT SIMILARITIES

Supply chains develop common sets of dominant motivators in order to find a niche in market environment. Agents who don't share them don't join or are the first which leave. The modal motivators and the key events that deliver motivation create a powerful framework of rules: rules that govern what constitutes smart behavior in a company. In our model the modal motivators are represented by the mental state of the agent or its internal strategic rule and the key events are considered the causes on which the agents react.

In the product space $S$ *$B$ of suppliers $S$ and buyers $B$, a match is a pair of two agents that represents the same business strategy. We analyze pairs of strategies according to some sort keys which represent the firm's output: product price variation, product quality variation or number of preferential customers. These criteria are called *matching variables* and they are used to identify matches. The matching variables are influenced by causes (firm's input), such as: component price variation, component quality variation, demand variation or political stability. A supply chain linkage decision rule is a rule that diagnoses a pair of strategies either as a link, a possible link or a nonlink. Matching weight or score (degree of similarity) is a number assigned to a pair that simplifies assignment of link and nonlink status via decision rules. Let $\gamma$ be an arbitrary agreement pattern in a comparison space $\Gamma$.

For instance $\Gamma$ might consist of $N$ patterns. Each represents a combination of causes' variation. $N$ is given by next formula: $N = b^c$. Here $b$ is called *base* and it represents the possible states of logic. We have considered a trivalent logic (b=3), because each cause and matching variable can increase, remain constant or decrease. Value $c$ is the number of causes considered. In our scenario we analyze situations with $c = 4$. We define also the next distances: 1) d(+,+) = d(0,0) = d(– ,–) = 0; 2) d(+,0) = d(0,+) = 1; 3) d(–,0) = d(0,–) = 1; 4) d(+,–) = d(–,+) = 2. Here d(+,+) means that the same matching variable $v$ increase at both agents at the same time. This is synonym with the fact that agents have the same strategy as regard to $v$. Hence, the distance between their strategies will be null from the $v$ point of view. In a similar fashion d(+,0) means that $v$ increase at the agent $B$ and remains unchanged at agent $S$ and d(+,–) means that $v$ increases at agent $B$ while it decreases at agent $S$.

*A. Supply chain formation*

We can already define a global distance $\Delta$ between two agents $B$ and $S$ on the entire space $\Gamma$:

$$\Delta = \frac{\sum_{i=1}^{N}\sum_{j=1}^{v} d_{ij}}{(b-1)\cdot v \cdot b^{c}} \tag{1}$$

This metric is a fraction from the maximum possible distance between two agents, computed according to all $N$ patterns and $v$ matching variables: $\Delta_{max} = (b - 1) \cdot v \cdot b^c$. Thus, $\Delta \leq 1$. Here $b$ - 1 comes from the fact that the largest distance is d (+, -) = d (-,+) = 2 = b-1. If we use a n-valent logic instead a trivalent one, this constant will increase correspondingly. The metric above supposes that all variables (price, quality and favorites clients) have the same weight in agent's decision. In practice, an agent is concerned that its partners to react similar as regard to its criterion. For instance, if it has a market strategy based on quality, it asks from its partners to act similar as regard to this criterion and it is more flexible with the others. Thus, the distance between each variable is weighted:

$$\Delta_{w} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{v} w_{ij}\cdot d_{ij}}{(b-1)\cdot b^{c}} \tag{2}$$

Due to the constraint _$wi$ = 1, the maximum distance has became $\Delta_{max} = (b\text{-}1)\cdot b^c$. The metric is computed using only constant distances $d$ defined above. But the agents perceive differently these distances and assign a different cost to each of them. To model this, we introduce a costs matrix $C_{b,v} = \alpha_{ij}$ where $\alpha_{ij}$ *in* $[1\text{-}\varepsilon, 1+\varepsilon]$. Depending on this cost, the distance $d_{ij}$ is perceived larger or not:

$$\Delta_{w,\alpha} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{v} w_{ij}\cdot \alpha_{ij}\cdot d_{ij}}{(b-1)\cdot b^{c}} \tag{3}$$

Until now, we have supposed that all $N$ patterns are uniformly distributed over the comparison space $\Gamma$, which is not quite true. For instance, a pattern of causes $\gamma$ = (+, +, 0, 0) means that the first two causes increase and the last two remain constant. Recall that, in our scenario, the causes are: components price variation, components quality variation, demand variation and political stability. It is more probable that a pattern $\gamma_1$ = (+, +, 0, 0) to happen in the future than $\gamma_2$ = (0, 0, 0, +). The last one tells that, while political risk increases or legal frame changes, the first three causes remain constant, which is not very probable. Therefore, we adapt the metric in order to manage such situations when probability of a pattern to appear $Pi$ is not constant over $\Gamma$:

$$\Delta_{w,\alpha,P} = \frac{\sum_{i=1}^{N} P_i \sum_{j=1}^{v} w_{ij}\cdot \alpha_{ij}\cdot d_{ij}}{b-1} \tag{4}$$

Due to the constraint $\Sigma P_i = 1$, the maximum distance has became $\Delta_{max} = b - 1$. When the variables weights and patterns probabilities are equal and distances are not influenced by costs, (4 ) becomes (1): $\Delta_{1/v,1,1/N}=\Delta$.

*B. Supply chain optimization*

According to the above metrics the agents found their similar partners and formed a supply chain. The metrics guarantee that the agents have quite similar strategies. Some differences between strategies have still remained. A way to overreach this situation through negotiation is described in [2]. The approach used there stops when the framework has found the differences between cognitive representations, and provide only some hints for human negotiation. We try to provide to the agents ability to find their self a common strategy. Our aim is to find the common strategy adequate for all the agents within the supply chain. In time, they have to adjust their strategy to this one in order to optimize the supply chain. We use a memetic approach. Here meme represents a unit of strategic information, analogous to the concept of gene in genetic algorithms.

We view these rules as co-adopted sets of memes, exactly as an organism's total chromosome is a coadapted set of genes. We consider each external strategic rule in our model as a *meme*. The memes replicate by modifying the agent mental state and consequently its business rules. We provide o framework in which we analyze how these units of strategic information propagate among the supply chain. The total number of memes in our model is $M = c \cdot b^v$. Each agent is an individual formed by $M$ memes. It assigns a weight to each meme. If it does not act on behalf of a rule, the correspondent meme has assigned a null weight. The fitness of the individual is its profit. The rules which have generated the biggest profit propagate more powerful through the supply chain. Putting in this way, the problem becomes how the agents adjust their rules' weights in order to converge to o single vector of weights. Hence, the supply chain evolves as a self organizing system. In firms, changes do not happen by the simultaneous introduction of new rules or by an abrupt adjust of the old ones. Fully 70% of corporate re-engineering programs fail. To avoid this alternative, the agents adjust their weights gradually, from generation through generation. In our scenario, the firms make public their profit once a month and this period define a generation. Due to simplicity reason we describe the memes propagation between two agents only. The weights for the $j$ meme at generation $i$+1 are adjusted recursively according to the next formula:

$$w_j^{i+1}(A) = w_j^i(A) \cdot f_A^i + w_j^i(B) \cdot f_B^i + w_j^i(A) \cdot f_B^i \qquad (5)$$

where $f_A^{\,i}$ represents the part of the profit obtained by the agent $A$ from total profit of the supply chain. The new weight of the agent $A$ is a sum of three terms. The first represents the level of confidence of agent $A$ in its own strategy. The second one describes how the agent $B$ perceives itself. Agents evolve defensive routines against new idea or new business rules. The last term models this reality: the $B$' profit is view through the eyes of $A$. It represents how the agent $A$ perceives agent $B$. In (5), if the agent $B$ has a zero profit it does not influence the rule propagation within the supply chain. In the section V we analyze an example describing how this approach works.

## V. EXPERIMENTAL RESULTS

*A. Supply chain formation*

In the case of a trivalent logic, the number of comparison patterns is $N = b^c = 3^4 = 81$. Consider that agent $B$ has to choose between suppliers $S1$, $S2$, $S3$, $S4$, with the following business rules[1].

**External strategic rules for agents $B$, $S1$, $S2$, $S3$, and $S4$**

| RULES agent $B$ | RULES agent $S1$ |
|---|---|
| priceIN(+)→priceOUT(+) | priceIN(+)→ qualityOUT(-) |
| demand(-)→favoriteClients(-) | demand( )→favoriteClients(+) |
| politicalStab(+)→qualityOut(+) | priceIn(+)→ priceOut(+) |
| qualityIn(+)→qualityOut(+) | politicalStab(0) →qualityOut(+) |

| RULES agent $S2$ | RULES agent $S3$ |
|---|---|
| priceIN(+)→qualityOUT(-) | priceIN(+)→qualityOUT(-) |
| demand(+)→favoriteClients(+) | demand(-)→ favoriteClients(+) |
| demand(+0→ priceOUT(-) | qualityIn(-)→qualityOut(-) |

| RULES agent $S4$ |
|---|
| priceIN(+)→ qualityOUT(+) |
| politicalStab(+)→ favoriteClients(+) |
| demand(+)→ priceOUT(-) |

Here, rules for matching variables that remain constant are not shown. For instance, some rules that are not mentioned: priceIN(-) . priceOUT(0) or priceIN(0) . priceOUT(0). We suppose that a set of past observations might be available from which the agent is able to infer some of the knowledge that are not public. For each potential pair ($B,S_i$) the algorithm observes how matching variables vary when causes are changing. Due to space considerations we have not depicted how decision depends on the chosen metric. In the above experiment we obtained that using Δ metric agent $B$ will perceive agent $S1$ to be the most similar, while using the other metrics agent $S4$ will be the favorite candidate. When the agent can obtain enough information from the market, the last three metrics provide a more precise solution. In fig. 2 we can see how the decision is changing when the causes are varying. It shows that if probability of the demand to increase is less than 0.5, the agent $S4$ has the most similar market strategy to the $B$ agent. When this probability is greater than 0.5 the agent $S1$ is the most similar to agent $B$.

*B. Supply chain optimization*

The total number of memes in our model is $M = c \cdot b^v = 4 \cdot 3^3 = 108$. We figured a situation with only 5 memes. Considering the next two agents: agent $A$ assigns the weights $v_A$(0.5, 0.3, 0.2, 0, 0) (meaning that it reacts only by the first 3 rules), while the agent $B$ $v_B$=(0.5, 0, 0.3, 0.2, 0). Suppose during 7 generations the agent $A$ obtained the

[1] We consider all the agents have a null or a positive profit; in a future work we will extend this approach to agents who might have negative balance.

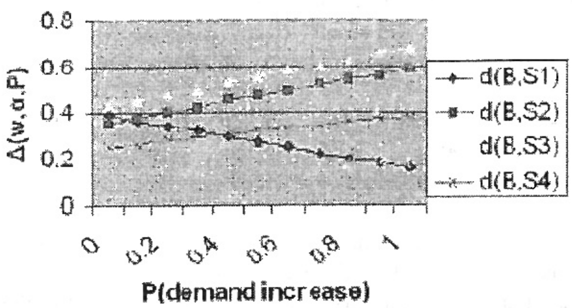

Fig. 2. The influence of the demand variation

next profits $H_A(50, 50, 50, 50, 50, 50, 50)$ while agent $B$ $H_B(20, 40, 60, 80, 100, 120, 140)$. Applying (5) we obtain fig. 3a. It depicts how memes' weights converge to the same value. Here, AM2 means the first meme of the agent A and BM2 the first meme of the agent B. In time these weights converge to the same value. Similar for pairs (AM3, BM3) and (AM4, BM4). Hence, all the memes will reach an equilibrium point. During this stage all the strategies evolve to a unique one. Thus, distances between them become zero. The result in fig. 3b bears this out: after 7 generations the supply chain reaches the unique strategy. The values provided might be useful hints in negotiation.

## VI. CONCLUSIONS AND FUTURE WORK

We have combined the advantages of causal representation, which is human readable, with the inference techniques from RuleML. The framework could be extended to deal with incomplete information, supposing that we don't now the entire comparison space Γ or even deal with conflicting rules, when the observed facts are noisy. At the moment, the agents deal with incomplete information by using a default system of rules. For instance, if information on price variation are missing, the agent will consider the default rule: *if priceIn(+) then priceOut(+)*. A further discussion in terms of privacy about rules could be also interesting. We provide also a method to find the unique strategy of a supply chain formed by agents with different strategies. The problem is to find the equilibrium point between strategies, equilibrium maintained by the amount of profits declared

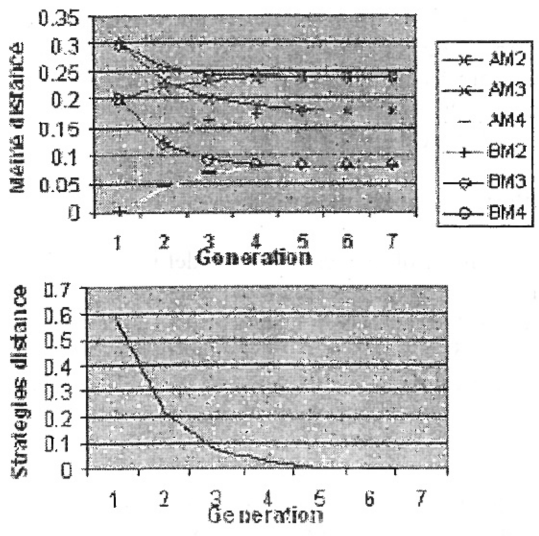

Fig. 3. a) Memes' convergence b) Equilibrium of the supply chain.

by each agent. Our future interest is to analyze more deeply this equilibrium in order to find some condition of convergence. It could be also interesting to study in what situation this equilibrium breaks and how the entire system adapts to the new changes.

## VII. ACKNOWLEDGMENT

This paper is created with support from the following projects: CNCSIS 423.

## VIII. REFERENCES

[1] Harold Boley, Said Tabet, and Gerd Wagner, "Design rationale of RuleML: A markup language for semantic web," Technical report, Semantic Web Working Symposium, 2001.

[2] Brahim Chaib-draa, "Causal maps: theory, implementation, and practical applications in multiagent environments," *IEEE Transactions on knowledge and data engineering*, 14(6):1201 1217, 2002.

[3] Andreas Eberhart, "An agent infrastructure based on the Semantic Web standards," In *Workshop on Business Agents and the Semantic Web*, 2002.

[4] Andreas Eberhart, "Ontoagent: A platform for the declarative specification of agents," In M. Schroeder and G. Wagner, editors, *International Workshop on Rule Markup Languages for Business Rules on the Semantic Web*, volume 60, Sardinia, Italy, 2002. COEUR.

[5] Benjamin N. Grosof, "Representing e-business rules for the semantic web: Situated courteous logic programs in RuleML," Technical report, MIT Sloan School of Management, 2002.

[6] Benjamin N. Grosof, Mahesh D. Gandhe, and Finin Timothy F, "SweetJess: Translating Daml- RuleML to Jess," In M. Schroeder and G. Wagner, editors, *International Workshop on Rule Markup Languages for Business Rules on the Semantic Web*, volume 60, Sardinia, Italia, 2002. COEUR.

[7] Benjamin N. Grosof and Terrence C. Poon, "Representing agent contracts with exception using XML rules, ontologies, and process descriptions," In M. Schroeder and G. Wagner, editors, *International Workshop on Rule Markup Languages for Business Rules on the Semantic Web*, volume 60, Sardinia, Italy, 2002. COEUR.

[8] Benjamin Olschok and Sabrina Scherer, "*Rule Applying Comparison Shopping Agent*," 2002. http://www.dfki.de/racsa.

[9] Jayashankar Swaminathan, Stephen Smith, and Norman Sadeh, "Modeling supply chain dynamics: A multiagent approach," *Decision Sciences*, 29(3):607–632, 1998.

[10] K. Taveter and G. Wagner, "Agent-oriented enterprise modeling based on business rules," In *Int. Conf. on Conceptual Modeling (ER2001)*, LNCS 2224. Springer- Verlag.

[11] David Trastour, Claudio Bartolini, and Chris Preist, "Semantic web support for the business-to-business ecommerce pre-contractual lifecycle," Technical report, HP Laboratories Bristol, 2003.