

USING AN ACTIVE FUZZY ECA RULE -BASED NEGOTIATION AGENT IN E-COMMERCE

Farnaz Mahan
Computer Science Department, Tabriz University
mahan@tabrizu.ac.ir

Ayaz Isazadeh
Computer Science Department, Tabriz University
isazadeh@tabrizu.ac.ir

Leili Mohammad Khanli
Computer Science Department, Tabriz University
l-khanli@tabrizu.ac.ir

ABSTRACT

E-commerce is considered a key service within modern information society, and the idea of automating e-commerce transactions has attracted much interest in recent years. A multi-agent model is a system that applies various autonomous agents to accomplish specified goals. Such a system addresses resource allocation issues. Because the nature of resource trading requires multiple agents to request geographically dispersed heterogeneous resources, we use a multi-agent architecture for e-commerce because each agent can be describe each participant intelligently. In this paper, negotiation agents based on fuzzy ECA rule-based proposed. Here we focus on agents in e-commerce that negotiate between sellers and buyers in order to get the best deal. The negotiation process between buyers and its sellers begins through combined and fairness protocols. We add learning properties to agents based on a fuzzy decision tree to develop negotiation skills and present the results. Using a fuzzy decision tree helps us understand and adapt other agents' behavior and real-time world conditions in order to produce the best contracts. Thus, the agent can improve in terms of skills on negotiation by updating its fuzzy decision tree.

Keywords: Electronic Commerce, Fuzzy Decision Tree, ECA Rule

1. INTRODUCTION

Today, achieving financial stability and economic growth and reducing poverty are discussed due to the importance of increasing market access and strengthening the rules-based trading system^{1,2}. As such, we need to learn more about quality management and understanding rules; further, we need advice on product development and adaptation negotiation. In recent years, developing e-commerce has shifted from a business establishing a simple Web presence to advanced uses of e-commerce technologies by improving its efficacy and profitability. The idea of automating e-commerce transactions, therefore, has attracted significant research interest. Most existing e-commerce systems involve human interventions, in which human beings make the most important decisions related to various activities. At the same time, some advocate that software agents are one of the best technologies for automating e-commerce processes. We anticipate that intelligent agents will be useful and flexible and that they could reduce the need for human intervention in crucial decisions^{3,4}. To adapt e-commerce to real-time environments, we use rule learning that has not been tested in previous work.

1.1 The Problem

This paper's aim is to investigate adaptive e-commerce negotiation agents related to real-time environments. The overall problem is to increase performance and subsequently profits from e-commerce activity. To meet this aim, we need a learning system to update the negotiation agent's rules. We need to recognize which rules should be consolidated, which rules should be modified, and which rules should be invalidated. Updating rules, therefore, must contain the ability to:

1. Change the value of some parameters in the rules.
2. Add new attributes to original rules and create new rules.
3. Create new rules that do not conflict with the original rules.

With learning, our resource management must have the following characteristics:

1. Increased responsibility of the negotiation agent in real time e-commerce.
2. Respond quickly with high performance and adaptive abilities.
3. Increase the profit of contracts accomplished through e-commerce.

1.2 Motivation

In previous works^{5, 6}, we presented experimental results using our own implementation negotiation mechanism in a model multi-agent e-commerce system with an agent-based negotiation method based on combined method and fairness. In this study, we used active database techniques to discover interesting behaviors within a dynamic environment and react intelligently to those environment changes.

An active database system (ADBS) is a database system that monitors a particular situation of interest and triggers an appropriate response in a timely manner when the interesting event occurs^{7, 8}. A system's active behavior is generally specified by means of rules, which describe the situations to be monitored and the system's reactions when these situations are encountered. In its general form, a rule consists of a rule event, a condition, and an action, known as Event-Condition-Action rules or ECA rules.

The negotiation model is composed of selling and buying agents, which negotiate in order to get the best deal. Here, we assume that agents are self-interested, the environment is dynamic, and both agents have deadlines. Such dynamism means that the agents' negotiation parameters (such as deadlines and reservation prices) are functions of both the state of the encounter and the environment. We are interested in endowing our agents with the flexibility necessary to engage in the unknown in advanced forms of negotiations using rule-based approaches. We start our presentation with active rule-based representations in automated negotiations. We suppose that the ECA rules help both the buyer and seller agent make decisions when events occurred¹.

In this paper, we want to learn about and update the ECA rules for e-commerce. Different representations of concepts may be learned from a set of labeled data set such as neural networks and decision trees^{9, 10}. Decision tree learning is reasonably fast and accurate. It is also straightforward in that it can reduce a decision tree to rules. The final representation used in this research consists of a rule base created from decision trees¹¹. A complimentary set of approaches uses decision trees to estimate probability, for example, in Classification and Regression Tree CART, which can also be viewed as a generalization of the classical logic^{12, 13, 14}. In our case, fuzzy logic captures the nature of human intuition better. The problem of branches missing from the decision tree is due to the fact that some of the reduced subsets at the non-leaf nodes do not necessarily contain examples of every possible value of the branching attribute¹⁵. Consequently, the decision tree may fail in some instances. In

this case, integrating decision trees with the knowledge component inherent in fuzzy sets is useful.

Important differences between crisp and fuzzy trees are represented in Figure.1¹⁰. While the general tree creates a piecewise constant approximation, fuzzy trees interpolate between the labels attached to their terminal nodes. Therefore, there are smooth input/output mappings, in the form of membership degrees¹⁶. This negotiation model can adapt the agent's strategy in response to the availability of resources and variations in negotiation parameters (deadlines and prices).

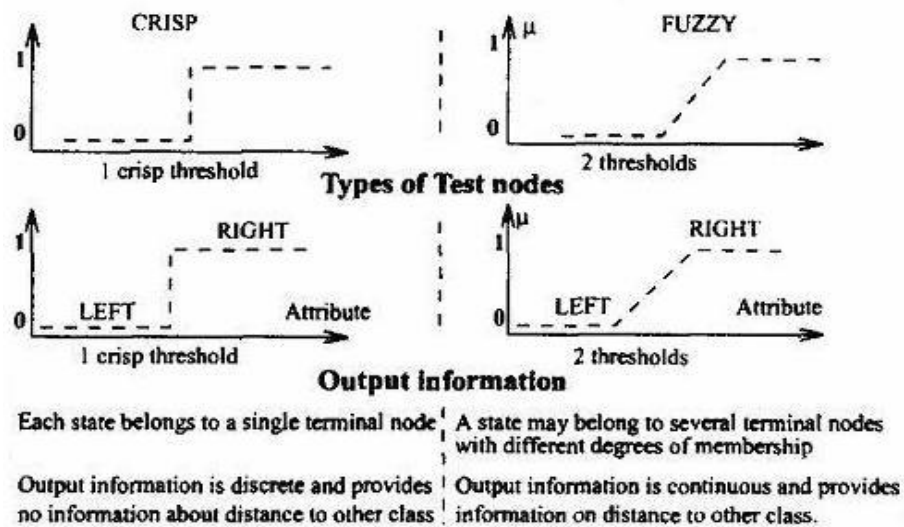


Figure 1. Main differences between crisp and fuzzy decision trees¹⁰

1.3 Objective and the Claim

This paper describes the research of a fuzzy decision that can be used to support flexible negotiation in e-commerce effectively. Our goal is to develop an updated active database decision system with new examples of conditions in a real environment base of events. The fuzzy decision tree, therefore, helps agents learn about a domain with the knowledge of each agent with changing membership. Some rules have static parameters that are determined by an expert, but when these rules use in a real-time environment, these parameters must be changed and updated based on data received to improve the performance of negotiation.

Because we receive instances at different times and under different conditions, we can also add new attributes such as time, load balancing, and profit to the rules. It helps that rule learning is efficient. The strategy pursued here is to break instances into n partitions based on events, then learn a decision tree for each of the n partitions in parallel^{17, 18}. At each learning period in each partition, we may have several fuzzy decision trees related to each rule; in this case, they must be combined in some way. In work by Chan and Stolfo, the decision trees are combined using meta-learning¹⁹. Further, in some partitions, it may be the case that some new fuzzy decision trees growing independently when there are new attributes for certain instances. So, the independent decision tree of each rule can be viewed as the agents learning a little more about a domain.

1.4 Paper Outline

This paper is organized as follows: In Section 2, we describe the background of rule-based negotiation and fuzzy decisions. In Section 3, we represent active fuzzy ECA rules and update them with using fuzzy decision trees. In Section 4, we evaluate our approach. Finally, we conclude the paper in Section 5.

2. BACKGROUND ON RULE-BASED NEGOTIATION

Rules have been indicated as a very promising technique for formalizing multi-agent negotiations³. We summarize the most important developments in the area of rule-based approaches to automated negotiations. Whenever an agent is submitted to negotiation, it also achieves a specification of the negotiation rules in terms of the shared ontology.

In several studies^{20, 21}, a mathematical characterization of auction rules proposed and parameterizing the auction design space is introduced that is organized along three axes: (a) bidding rules, which state when bids can be posted, updated, or withdrawn; (b) clearing policy, which states how the auction allocates resources (including auctioned items and money) between auction participants (this corresponds roughly to agreement making in our approach); and (c) information revelation policy, which states how and what intermediate auction information is supplied to the participating agents.

Reeves et al.,²² developed a special declarative language, called CLP (“Courteous Logic Programs as KR (knowledge representation)”) to express and reason situations related to contracts and negotiations. They

also showed how rules generated during the negotiation process can be combined with the partial contract to form an executable final contract. This project was a continuation of the Michigan AuctionBot project ³.

Background knowledge supporting this infrastructure was defined in three CLP rule sets:

Auction-Configuration

Auction-Space

Auctionbot-Mapping. Auction-Configuration, in three CLP rule sets which supports reasoning about alternative negotiation structures.

A new rule-based scripting language, AB3D, for expressing auction mechanisms, Lochner and Wellman reported in²³. The design and implementation of AB3D were considered to parametrization in^{20, 21} as were experiences of the Michigan Internet AuctionBot. A3BD initializes auction parameters, defines the rules for triggering auction events, declares user variables, and defines the rules for controlling bid admissibility²³.

Using defeasible logic programs is reported in²⁴ that provide a formal executable approach for defining the strategy of agents participating in negotiations. Also another researchers¹⁸ describe a defeasible logic to implement a system of agents that negotiate with strategies. Note, too, that defeasible logic programs can express logic programs proposed in²² and also to support efficient reasoning. This suggests that they might be the appropriate formal representation of negotiation strategies.

The CONSENSUS system was presented in²⁵. In CONSENSUS, each agent uses a rule base partitioned into three parts: (a) basic rules for negotiation protocol; (b) strategy rules for negotiation strategy; and (c) coordination rules that determine the knowledge for assuring that either all of the complementary items or none of them are purchased.

Another memorable work for automated negotiations is specifically targeted to auctions (ARM¹⁹) and its associated declarative auction specification language (DAL¹⁷). DAL constructs contain the following: views, validations, transitions, and agreement generators. Views are for visibility rules; validations are for validity and protocol enforcement rules; transitions update rules; and agreement generators are used to form agreement and negotiation life-cycle rules.

In previous works^{5, 6}, we presented the experimental results of our own implementation negotiation mechanism in a model named “multi-agent system e-commerce,” which features an agent-based negotiation method based on combined method and fairness. In⁵ we used a Genetic algorithm to determine the best agents to optimize the system of negotiation. This negotiation model can adapt the agent’s strategy in response to available resources and variations in negotiation parameters (deadlines and prices). In⁶, an agent-based negotiation method based on Adaptive Fuzzy in bilateral contracts of energy was proposed. Further, we developed a model suitable for dynamic electricity market environments. Specifically, we used Adaptive Fuzzy to optimize the system of negotiation. And, we discussed ECA rule-based approaches for automated negotiation in a model of our multi-agent ecommerce systems in¹. Our discussion was supplemented with experimental results obtained using our own implementation of an ECA rule-based price automated negotiation framework. The results support the claim that ECA rules are a feasible and scalable technology for approaching flexible automated negotiation in e-commerce¹.

In this study, we used active database techniques composed with fuzzy to discover interesting behaviors of a dynamic environment and react intelligently to the environment changes.

Developing and using Fuzzy set theory has grown widely and various generalizations have been applied. The concept of fuzzy rules is one of these generalizations. Fuzzy rules have been applied in different areas such as logic programming and decision-making problems²⁶. In this paper, we study fuzzy decision ECA rules.

Decision trees classify data from the root to leaf nodes, while the fuzzy decision tree classifies data to multiple branches of a node with different satisfaction degrees ranged on [0,1]. Fuzzy sets used to build the tree are imposed on the algorithm. One of main objectives of the fuzzy decision tree is to bring a high degree of accuracy of classification for unknown cases²⁶. Experimental results in^{10, 26} showed this can be achieved.

The fuzzy decision tree approach is designed for classification problems with attributes and classes represented in fuzzy linguistic terms²² because the attribute values of training data are fuzzy. Accordingly, fuzzy representation performs very well when dealing with problems of uncertainty, noise, and inexact data. The fuzzy decision tree has been used for handling fuzzy decision problem-solving problems successfully²⁶.

3. ACTIVE FUZZY ECA RULE NEGOTIATION AGENT IN E-COMMERCE

E-commerce (EC) is “a modern business methodology that addresses the needs of organizations, merchants, and consumers to cut costs while improving the quality of goods and services and increasing the speed of service delivery. The term also applies to the use of computer networks to search and retrieve information in support of human and corporate decision-making²⁷.” Indeed, e-commerce has been adopted widely in most enterprises. Yet, we need to optimize our rules dynamically and learn new rules historically.

The basic potential for interactive data processing is to provide an immediate situation-specific response, delivery problem-specific advice instantly, and develop better ways to access and inspect the supplier’s offer²⁸.

A multi-agent model is a useful system to accomplish some specified goals²⁹. We can use multi-agent systems simply because the nature of resource trading requires multiple agents to request geographically dispersed heterogeneous resources. To support such a dynamic situation, a multi-agent system must be extended to develop an adaptive agent with a high ability to negotiate. Such a system must allow the agent to learn agent during interactions and from the requirements of the resources advertised³⁰.

Recently, developing, implementing and experimenting with a multi-agent e-commerce system have been discussed³¹. In our work, we want to provide agents with the flexibility required for negotiations. In this context, rule-based approaches have been developed as a very useful technique for parameterizing the negotiation³¹. It provides rules for describing either negotiation strategies, mechanisms, or both³¹. In designing systems for automated negotiations, we need to recognize different negotiation mechanisms. We also must define the rules between participants and the negotiation strategies that these rules, which provide behaviors in order for an agent to achieve a desired outcome. The result of negotiation depends on (a) the resources available for negotiation; (b) the negotiation parameters (e.g., deadlines, delivery time, and prices); and (c) the negotiation strategies of the agents. All three of these items are all subject to change.

We thus need a model in which the agents select strategies at each stage of the negotiation based on the current state and other agent states available to achieve its objectives (e.g., reaching an agreement before a

deadline). We believe that effective strategies can be chosen based only on the current state of the system and must be independent of the history of the negotiation process. We thus propose using the ECA rule-based system to create active behaviors for agents in e-commerce (see Figure 2).

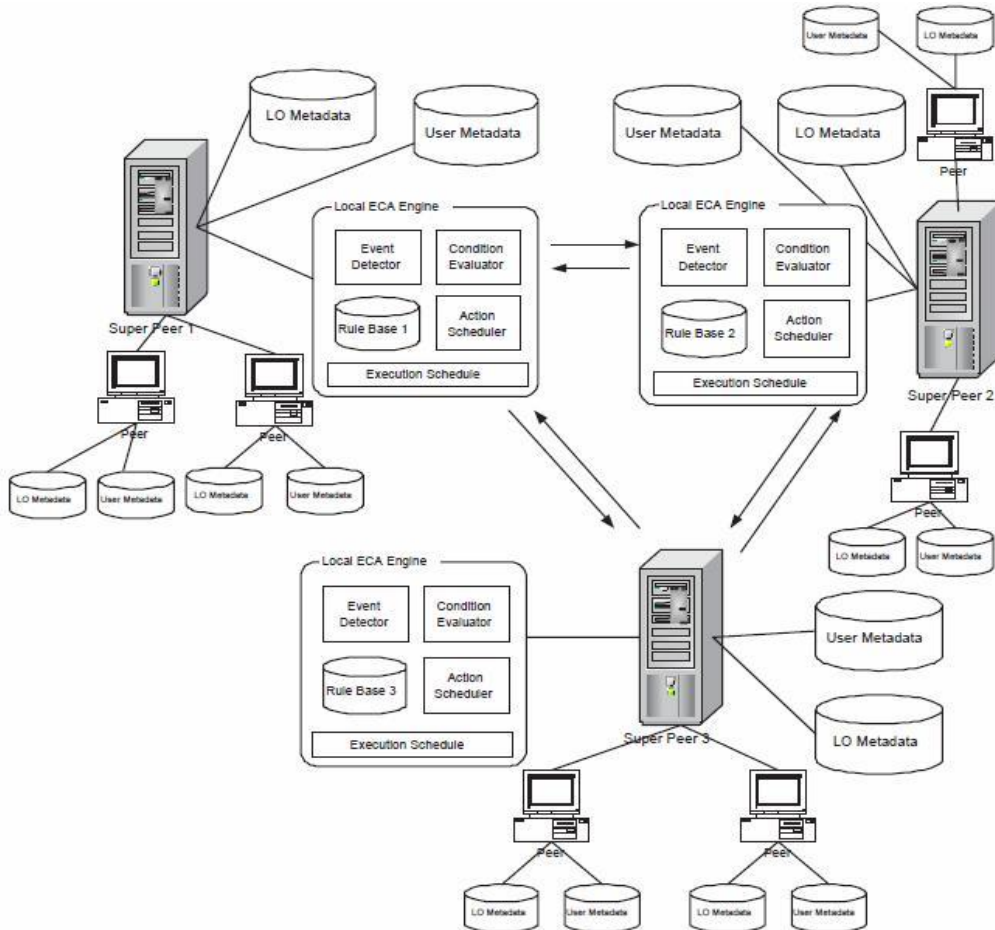


Figure 2. Distributed ECA rule-based multi-agent system

3.1 The Structure of Seller and Buyer Agents

Buying and selling agents have similar structures and a kind of symmetrical behaviour due to their antagonistic objectives. The structure of agents contains three functional modules: (a) Events Handler, (b) Negotiation Management, and (c) ECA rule-based Decision Making in addition to one ECA rule-based module: the Market and Individual Knowledge module. We changed the structure to ECA rule-based

knowledge used in our previous research studies in^{5,6}. Figure 3 illustrates this structure.



Figure 3. Seller and Buyer Agents Structure

3.2 The Combined Negotiation Method and Agent Behaviors

The negotiation process used in this paper is a combined method⁸. This method includes three main strategies for negotiation: (a) Time-Dependent strategy; (b) Behavior-Dependent strategy; and (c) Resource-Dependent strategy. Each of strategies itself can include different kinds of sub-strategies; thus, there can be m strategies in the negotiation of agents. In the combined negotiation method, we can define minimum and maximum values for each of the negotiation subjects. These minimum and maximum values are defined based on the previous experiences of the agents as well as their objective functions. We characterize the main components of our model in⁵.

When considering the design of systems for automated negotiations, it is typically the case that negotiation protocols (or mechanisms) that define “rules of encounter” between participants and negotiation strategies that define behaviors aiming at achieving a desired outcome are distinguished. However, rule representations were proposed for both negotiation mechanisms and strategies^{3,4}. Whenever an agent is submitted to negotiation, it also achieves a specification of the negotiation rules in terms of the shared ontology.

Negotiation rules are used to manage the negotiation mechanism. Rules are organized into several parts: “rules for participants' admission to

negotiations, rules for checking the validity of proposals, rules for protocol enforcement, rules for updating the negotiation status and informing participants, rules for agreement formation and rules for controlling the negotiation termination³².

3.3 ECA Rule in E-commerce

Active database systems are distinguished by their ability to monitor predefined situations automatically and react to them based on predefined actions. The reactive semantics must be centrally managed on a timely basis. An active database system must have the ability to provide a knowledge model and an execution model to support the reactive behavior^{32, 33}.

ECA rules are one way to implement this kind of functionality^{7, 8}. An ECA rule has the general syntax as follows:

DEFINE RULE rule_name

ON event

IF condition

DO action

The rule_name identifies the rule that is unique. A rule description commonly includes three components:

1. Events: The event part of a rule describes a happening to which the rule may be able to respond. Events can be roughly subdivided into two categories: primitive events that correspond to elementary occurrences and composite events that are composed of other composite or primitive events.
2. Conditions: The condition part of the rule specifies what has to be checked once the rule is triggered but before it is executed. If the result of the condition evaluation is true, the condition is satisfied.
3. Actions: Actions describe the task the rule considers relevant to the event and the condition.

There are several advantages in using ECA rules to implement this kind of functionality compared to directly implementing application code³³:

- ECA rules allow an application's reactive functionality to be specified and managed within a rule base rather than being encoded in diverse programs, thus enhancing the modularity, maintainability, and extensibility of applications.
- ECA rules have a high-level, declarative syntax and are thus amenable to analysis and optimization techniques that cannot be easily applied if the same functionality is expressed directly in application code.
- ECA rules are a generic mechanism that can abstract a wide variety of reactive behaviors. In contrast, application code is typically specialized to a particular kind of reactive scenario.

One of the languages that has used ECA rules on the web is ECA-ML language. Users register rules in the ECA-ML language³⁴ at an ECA service on the web that provides the infrastructure and global rule semantics. Figure 4 illustrates the ECA Rule Components and Corresponding Languages³⁵. The global architecture and the general markup principles are described in³⁴.

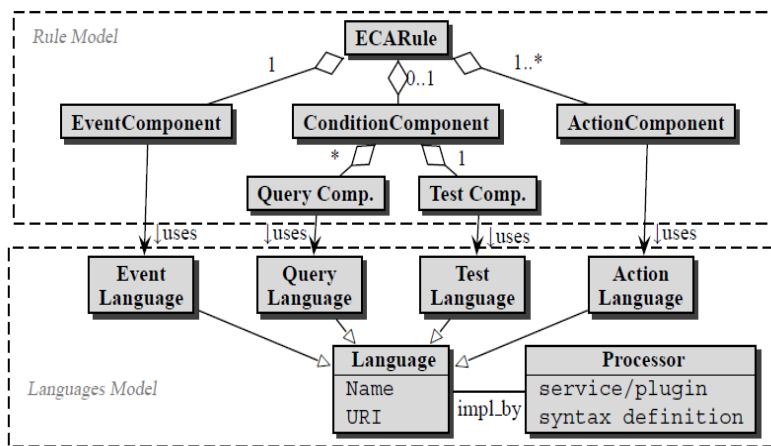


Figure 4. ECA Rule Components and Corresponding Languages³⁴

When events occur, all agents can be active via its ECA rule base, such that the system can compute and evaluate the set of possible candidate objects that could be presented to the user. Computation of this set and its associated value to the buyer and seller, however, is only a means to determine the next needs-oriented question that will optimize the expected focus-set value.

For the purpose of this paper, we set up our system for a particular negotiation scenario base of English auctions rules. The rules were then used to initialize rule inference engines encapsulated by the negotiation hosts^{3, 4}. Let us now consider a few samples our ECA rules base on English auctions rules:

INFORM-SELLER rule specifies that a buyer participant can post a proposal whenever there is an offer already posted by a seller participant.

INFORM-SELLER rule

ON POSTING-BUYER

IF

There is a valid proposal, Pr, of a participant with role buyer \wedge There is an active proposal of a participant with role seller

THEN

Proposal Pr is posted

AGREEMENT-FORMATION rule specifies that whenever agreement formation is triggered, if the currently highest bid is greater than the seller reservation price (that it is not disclosed to the participants), an agreement is formed between the submitter of the highest bid and the seller.

AGREEMENT-FORMATION rule

ON SUBMIT BID

IF

The currently highest bid is B and was submitted by buyer S1 \wedge There is an active proposal of seller S2 with price P \wedge Negotiation is on goods A \wedge B \geq P

THEN

An agreement of S1 with S2 to transact goods, A, at price, P1, is formed

3.4 Fuzzy Decision Tree for ECA Rule Update

A fuzzy decision tree is composed from fuzzy sets and symbolic decision trees. This composition improves the representative power of decision trees with the knowledge component in fuzzy logic, leading to more robustness (noise immunity) and applicability in imprecise

contexts¹⁴. A fuzzy decision tree divides each attribute into fuzzy sets and assigns membership degrees to values of attributes according to membership functions. In actuality, every edge is annotated with a condition, and every leaf is annotated with a fuzzy set over relevant class (see e.g., Figure 5). For conciseness, we consider only binary trees in this paper, where one of the conditions at the outgoing edges of a node is chosen from τ (the test condition of the node), and the condition at the other outgoing edge is the negation of the test condition^{13, 14}. A further restriction is that each condition is used at most once in each path from root to leaf¹³.

Whenever the conditions with which the edges of a path are annotated are true for an example, we expect the example to belong to the relevant classes according to the fuzzy set of the leaf annotated with it. Note that the conditions on the outgoing edges of a node are not necessarily reciprocally exclusive. Therefore, we have to combine clues given by all paths into one final result. We used a method¹³ that calculates the weighted average of the fuzzy sets with which leafs are annotated, where each fuzzy set is weighted by the degree the example belongs to the conjunction of the conditions on the edges of the path. One important property is that if a crisp output value is desired, one can apply any of the well-known defuzzification methods; for example, taking the class with the highest truth value¹³. In this paper, we focus on rule learning that uses the fuzzy decision tree to update rules.

When active semantics are changing, ECA rules have to be updated by considering the new semantics and conditions. Updating rules is absolutely necessary for the rule set to remain useful. To achieve this update and learning, we used the fuzzy decision tree for each rule-base of the event.

Here we want to update the ECA rules for high performance as negotiation agents in e-commerce. We create initial fuzzy decision trees for each rule of base ECA rules that we suppose the root node of them is the related event of each rule. At the end of each tree, we have a maximum of two leafs that describe the actions of each rule. When the rule event occurs, it refers to base fuzzy ECA rule related to it in agent. If it adapts with the conditions of related fuzzy rule, then does not need to update. But if it does not adapt or provide better and faster negotiation techniques, we need to update the rule so fuzzy decision tree is useful.

4. CASE STUDY

Under the model explored in this paper, a training data set is new condition that occurs in a real environment. In agents, a training data set will be broken into n subsets with regard to events. Fuzzy decision trees learn from each of these n subsets in parallel. For example, one of original rule is:

AGREEMENT-FORMATION rule

ON SUBMIT BID

IF

The currently highest bid is B and was submitted by buyer S1 ^ There is an active proposal of seller S2 with price P ^ Negotiation is on goods A ^ B >=P

THEN

An agreement of S1 with S2 to transact goods A at price P1 is formed. In general, the fuzzy decision tree of this rule is shown in Figure 5. P is a continuous attribute that starts when the expert determines it and then may need to change based on examples and new data that is received from the environment. In an actual case, when the number of refers to “not do anything” are increasing, we need learning and changing the value of parameter such P to occur, so that the value of P is not static and must be learning all the time. Fuzzy decision trees for learning are included in the update rule step.

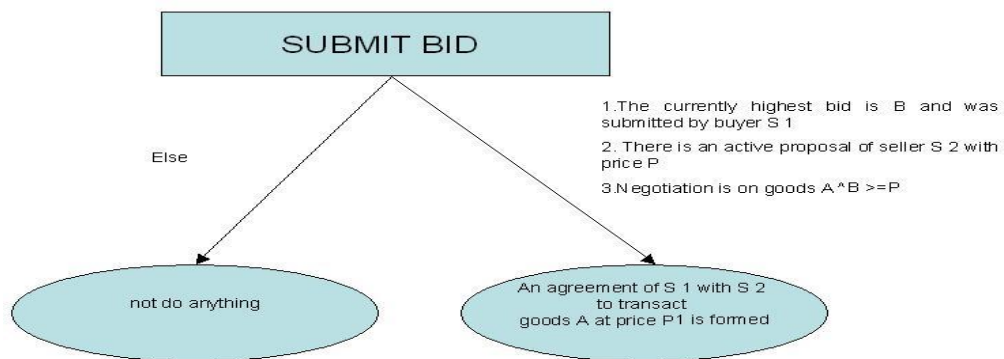


Figure 5. Fuzzy decision tree of example ECA rule

4.1 Experimentation via Simulation

We have evaluated the performance of the resource management ECA rules strategies with using the fuzzy decision tree. We have shown that the ECA rules based on the fuzzy decision tree can improve negotiation throughput.

In using ECA rules for e-commerce, when we received an event with a different attribute value, we could not respond about to about 40% of these cases because the value of threshold parameters was static. Hence, when our threshold parameters change with fuzzy degree and use fuzzy decision trees for managing events, our agents could respond to more than 93% of the events successfully and quickly. Figures 6 and 7 represent the results of our approach.

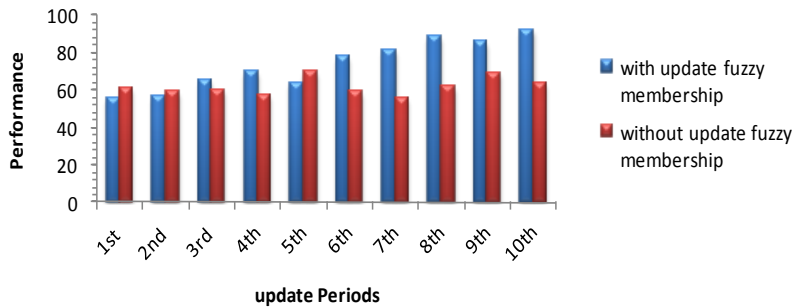


Figure 6. Compare performance of systems

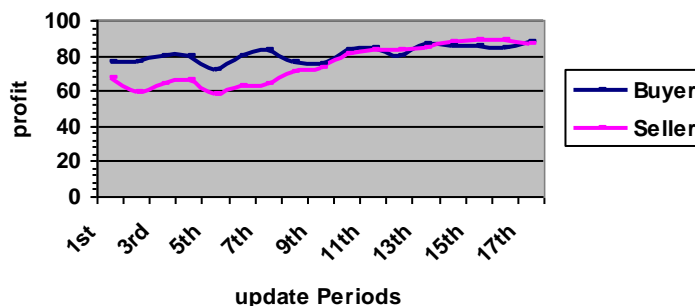


Figure 7. Average of profit for seller and buyer

5. CONCLUSION

E-commerce is one of the important services available in today's information society; therefore, the ability of software agents to discover remote markets and employ commercial negotiations management based on unknown market mechanisms is of primary importance. Today, a key aspect for the success of automated negotiations is that it has already generated some interest in the research. Rules establish a very useful approach to defining negotiation processes, and open rule-based semantic descriptions of the market mechanisms are developed.

In this paper, we have discussed ECA rule-based approaches for automated negotiation in a model of multi-agent e-commerce systems. Our discussion was included experimental results obtained using our own implementation of an ECA rule-based price automated negotiation framework. The results support the claim that ECA rules are a feasible and scalable technology to approach flexible automated negotiation in e-commerce. For future work, we plan to integrate ECA rule learning into our agent-based model e-commerce system. This will be based on the fuzzy decision tree approach to learning new examples and data as well as provide dependability and efficiency because some parameter must be learned. This negotiation model can adapt the agent's strategy in response to resource availability and variations in negotiation parameters (deadlines and prices).

5.1 Demonstration of the Claim

We now return to Section 1.1, where we defined the overall problem this paper addressed. Recall that our proposed solution, ideally characterized as a method, satisfied specific characteristics that we presented in Section 3.4.

1. In Section 3.4 we presented a fuzzy decision tree with rule update satisfying the first characteristic of the solution.
2. For the second and third characteristics, we evaluated our work in section 4 and presented how our approach responded quickly and had high performance and an adaptive nature without a learning system. Results are represented in Figures 6 and 7.

5.2 Future Work

Several directions can be explored to extend and improve this work. We can use flexible fuzzy decision trees to best adapt the rules to dynamic

e-commerce environments. Also the nero-fuzzy approach can be used for best learning for ECA rules-based negotiation agents. For future work, it is also possible to update or invalidate the existing rules by mining the triggered rules history.

6. REFERENCES

- [1] Farnaz Mahan, Ayaz Isazadeh, and Leili Mohammad Khanli, ECA Rule-based negotiation agent in e-commerce. *Proceedings of the 2010 International Conference on Communication Software and Networks (ICCSN 2010)*, February 26-28, Singapore, 2010.
- [2] Hiroyuki Kawano, Sho jiro Nishio, Jiawei Han, and Toshiharu Hasegawa, How Does Knowledge Discovery Cooperate with Active Database Techniques in Controlling Dynamic Environment. *Proceedings of the 5th International Conference on Database and Expert Systems Applications*, September 7-9, Greece, 1994.
- [3] Costin B̃adic̃a, Maria Ganzha, and Marcin Paprzycki, Rule-Based Automated Price Negotiation: Overview and Experiment. *Lecture Notes in Computer Science*, LNCS 4029, p1050-1059, 2006.
- [4] Costin B̃adic̃a, Adriana B̃adit̃al, Maria Ganzha, Alin Iordache, and Marcin Paprzycki, Rule-Based Framework for Automated Negotiation:Initial Implementation. *Proceedings of the RuleML'2005*, November 10-12, Ireland, 2005.
- [5] Farnaz Mahan, and Mohammadreza Kangavari, Using Multi-Agents Model in Development of Negotiation Based on Combined Method and Fairness in Bilateral Contracts for Electricity Market. *Proceedings of the IEEE International Conference on Mechatronics and Automation*, August 5-8, China, 2007.
- [6] Farnaz Mahan, Mohammadreza Kangavari, and Sahar Mamizadeh, Using Adaptive Fuzzy Negotiation of Multi-Agents Model in Bilateral Contracts for Electricity Market. *Proceedings of the International Conference on Future Computer and Communication*, April 3-5, Malaysia, 2009.
- [7] L.M. Khanli, M. Analoui, and Grid-JQA, Grid Java based quality of service management by Active database. *Proceedings of the 4th Australian Symposium on Grid Computing and e-Research, AusGrid*, January 16-19, Australia, 2006.
- [8] Min Dai, Ya-Lou Huang, Data Mining Used in Rule Design for Active Database Systems. *Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery*, August 24-27, China, 2007.

- [9] J. J. Grefenstette, C. L. Ramsey, and A. C. Schultz, Learning sequential decision rules using simulation models and competition. *Machine Learning*, 5(4), p355-381, 1990.
- [10] Lawrence O. Hall, Nitesh Chawla, and Kevin W. Bowyer, Decision Tree Learning on Very Large Data Sets. *Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics*, October 11-14, San Diego, CA, USA, 1998.
- [11] E. Frank, and K. Huber, Active Learning of Soft Rules for System Modelling. *Proceedings of the 2nd European Congress on Intelligent Techniques and Soft Computing*, September 20-23, Aachen, Germany, 1996.
- [12] Dae-Ki Kang, Kiwook Sohn, Learning decision trees with taxonomy of propositionalized attributes. *Pattern Recognition*, 42(1), p84-92, 2009.
- [13] Marina Guetova, Steffen Hölldobler, and Hans-Peter Störr, Incremental Fuzzy Decision Trees. *Proceedings of the 25th Annual German Conference on AI: Advances in Artificial Intelligence*, September 16-20, Aachen, Germany, 2002.
- [14] Sattar Hashemi, and Ying Yang, Flexible decision tree for data stream classification in the presence of concept change, noise, and missing values. *Data Mining and Knowledge Discovery*, 19(1), p95-131, 2009.
- [15] Ding-An Chiang, Wei Chen, Yi-Fanwang, and Lain-Jinn Hwang, Rules Generation from the Decision Tree. *Journal of Information Science and Engineering*, 17(2), p325-339, 2001.
- [16] Xavier Boyen, and Louis Wehenkel, Automatic induction of fuzzy decision trees and its application to power system security assessment. *Fuzzy Sets and Systems Journal*, 102 (1), p 3-19, 1999.
- [17] D. Rolli, and A. Eberhart, An Auction Reference Model for Describing and Running Auctions. *Proceedings of the 7th International Conference Wirtschaftsinformatik*, February 23-25, Bamberg, Germany, 2005.
- [18] T. Skylogiannis, G. Antoniou, and N. Bassiliades, A System for Automated Agent Negotiation with Defeasible Logic-Based Strategies - Preliminary Report. *Proceedings of the Rule ML'04*, November 8, Japan, 2004.
- [19] D. Rolli, S. Luckner, A. Gimpel, and C. Weinhardt, A Descriptive Auction Language. *Electronic Markets*, 16(1), p51-62, 2006.
- [20] P.R. Wurman, M.P. Wellman, and W.E. Walsh, A Parameterization of the Auction Design Space. *In Games and Economic Behavior*, 35(1/2), p 271-303, 2001.
- [21] P.R. Wurman, M.P. Wellman, and W.E. Walsh, Specifying Rules for Electronic Auctions. *In AI Magazine*, 23(3), p15-23, 2002.

- [22] D. M. Reeves, B. N. Grosz, M. P. Wellman, and H. Y. Chan, Toward a declarative language for negotiating executable contracts. *Proceedings of the AAAI-99 Workshop on Artificial Intelligence in Electronic Commerce (AIEC-99)*, July 18-19, Orlando, Florida, 1998.
- [23] K.M. Lochner, and M.P. Wellman, Rule-Based Specification of Auction Mechanisms. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, July 19-23, USA, 2004.
- [24] G. Governatori, M. Dumas, A.H.M. ter Hofstede, and P. Oaks, A formal approach to protocols and strategies for (legal) negotiation, *Proceeding of the 8th International Conference on Artificial Intelligence and Law*, May 21-25, USA, p168-177, 2001.
- [25] Benyoucef, H. Alj, K. Levy, R.K. Keller, A Rule-Driven Approach for Defining the Behaviour of Negotiating Software Agents. *Proceedings of the DCW'2002*, April 3-5, Australia, 2002.
- [26] Rui-Yang Chen, A problem-solving approach to product design using decision tree induction based on intuitionistic fuzzy. *European Journal of Operational Research*, 196(1), p266-272, 2009.
- [27] E.W.T. Ngai, and F.K.T. Wat, Fuzzy decision support system for risk analysis in e-commerce development. *Journal of Decision Support Systems*, 40(2), p235-255, 2005.
- [28] Ulrich Thiel, and Adelheit Stein, *Intelligent E-Commerce with Guiding Agents based on Personalized Interaction Tools, E-Business: Key Issues, Applications and Technologies*. NL: IOS Press, Amsterdam, 2000.
- [29] Cheng Wai Khuen, Chan Huah Yong, and Fazilah Haron, A Framework for Multi-Agent Negotiation System Using Adaptive Fuzzy Logic in Resource Allocation. *International Journal of Information Technology*, 11(4), p35-49, 2005.
- [30] Costin Bădică, Adriana Bădită, and Maria Ganzha, Implementing rule-based mechanisms for agent-based price negotiations. *Proceedings of the 2006 ACM symposium on applied computing*, April 23 -27, France, 2006.
- [31] Charles M. Macal, and Michael J. North, Tutorial on Agent-Based Modeling and Simulation. *Proceedings of the 37th Winter Simulation Conference*, December 4-7, USA, 2005.
- [32] J. Demsar, Statistical comparison of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7, p1-30, 2006.
- [33] M. Zoumboulakis, G. Roussos, and A. Poulouvasilis, Active Rules for Sensor Databases. *Proceedings of the First Workshop on Data Management for Sensor Networks (DMSN)*, August 30, Canada, 2004.

- [34] W. May, J. J. Alferes, and R. Amador, *Active rules in the Semantic Web: Dealing with language heterogeneity. International conference of Rules and Rule Markup Languages for the Semantic Web*, November 10-12, Ireland, 2005.
- [35] Erik Behrends, Oliver Fritzen, Wolfgang May, Franz Schenk, and Daniel Schubert, A Framework and Components for ECA Rules in the Web (Demo). *Proceedings of the International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML 2006)*, November 10-11, USA, 2006.

