Negotiating With Experience

Wai Yat Wong, Dong Mei Zhang, Mustapha Kara-Ali

CSIRO Mathematical and Information Sciences
Locked Bag 17, North Ryde NSW 2113 Australia
[wai.wong, dong.mei.zhang]@cmis.csiro.au

Abstract

Current E-Commerce trading systems which look at electronic negotiation facilities usually uses predefined and non-adaptive negotiation mechanisms [Maes, et. al. 1999]. This paper presents a negotiation framework which applies Case-Based Reasoning(CBR) techniques to capture and re-use previously successful negotiation experiences. This experience based negotiation framework provides adaptive negotiation strategies that can be generated dynamically and are context-sensitive. We illustrate this framework in the used car trading domain and have developed a web-based negotiation module for used-car trading. This paper begins by describing our formulation for negotiation, our CBR approach within negotiation framework and our representation of each negotiation case pertaining to the used car trading domain. It focuses on our methodologies to match and re-use similar negotiation experience. It also presents the software architecture, design and implementation of our web-based used-car trading module. We believe that our experience based negotiation framework will enhance the negotiation skills and performance of current trading agents.

Introduction

Automated negotiation is becoming an integral and important part of E-Commerce. Real-world negotiations in general accrue transaction costs and times that may be too high for either consumers and merchants [Maes et .al. 99]. The benefit of a good automated negotiation mechanism is well recognized [Sandholm 99]. A good automated negotiation can both save time and find better deals in the current complex and uncertain E-Commerce environment.

Most current e-commerce systems use predefined and non-adaptive negotiation strategies in the generation of offers and counter-offers during the course of negotiation. For example, negotiation in Kasbah [Chavez et .al. 97, Maes et. al, 99] (MIT media Lab's) uses three predefined strategies, anxious, cool-headed and frugal corresponding to linear, quadratic and exponential functions in the generation of proposals/counter-proposals. Buyers/sellers themselves have to decide which strategy to take before the negotiation starts. Researchers are now exploring various Artificial Intelligence based techniques to provide adaptive behavior in the negotiation engine. The use of Bayesian learning to learn negotiation strategy is one example [Siriwan et. al. 99, Zeng et.al. 98].

Good negotiation skill in humans seems to come from experience. This observation has motivated us to focus on Case Base Reasoning (CBR) as an approach to use past negotiation experience/strategies as guides to suggest suitable strategies to the current negotiation situation. We are currently examining the effectiveness of this Experienced Based Negotiation framework in the car trading domain [Kowalczyk R et .al. 99] to produce better and efficient negotiation.

This paper introduces the software components of our Experienced Based Negotiator system, our definition of negotiation strategies. It presents a concise overview of our experience based negotiation framework. It summarizes the car trading domain where we deploy our negotiation framework and presents the representation of our negotiation case base. It focuses on our methodologies to match and re-use similar negotiation experience. It also presents the software architecture, design and implementation of our system.

Components of Experienced Based Negotiator

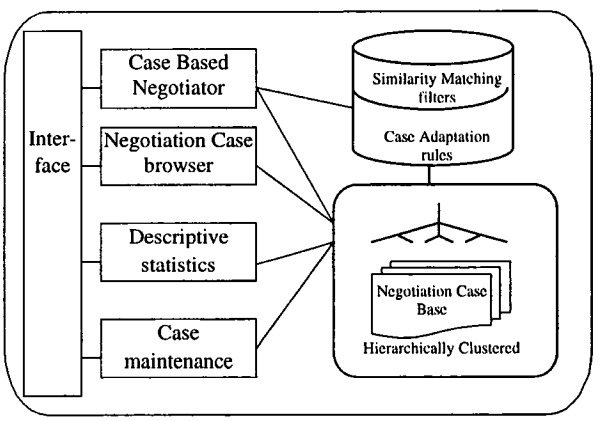

Figure 1: Components of negotiator prototype

Our Experience Based negotiator prototype contains several functional components as shown in Figure 1, a Case-Based Negotiator, a Case Browser, a Statistics component and a Case maintenance component:

• The Case-Based Negotiator assists the users to negotiate with opponent agents on used car trading price. It matches current negotiating scenario with previous successful negotiation cases and provides appropriate counter-offers for the user based on the best matched negotiation case. Resultant successful negotiation can be automatically updated into the negotiation case repository.

• The Case Browser allows users to browse a previous negotiation case repository using various queries.

• The Statistics component supplies several useful descriptive statistics on negotiation case repository.

• Finally, the Case Maintenance component allows negotiation experts to moderate, maintain and to update the case repository. To condense potential great number of similar stored cases, generalization based /concept learning techniques can be used here to induce generalized case. This will allow efficient, faster and perhaps better case matching and reuse in the future.

A user of the Experience Based Negotiator starts by selecting choice of used car to buy/sell. The user then input profile information, e.g. negotiation focus (good-deal, best-price, must-buy/sell), budget, sex, age, etc. The negotiator will start negotiation with the opponent agent once the user is satisfied with the input and selection. During a decision-making moment in the negotiation process, the case base negotiator retrieves relevant cases from the current negotiating scenario and adapts the episodic strategy from the best-matched negotiation case to generate a counter-offer. The dynamics of the negotiation process will be tracked and displayed on the user screen (figure 2).

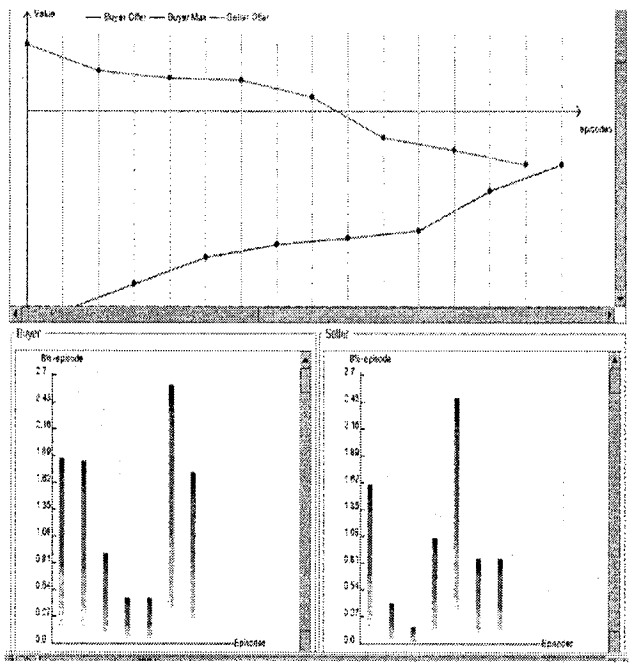

Figure 2: Tracking the negotiation process

Formulation of negotiation process

Negotiation is an iterative and sometimes lengthy process [Sycara, 93] where the trading agents start by having goals that may be far apart and whose distance has to be narrowed gradually. A successful negotiation occurs when the two opposing goals meet, e.g.. when the buyer's price meets the seller's offer or vice versa. So, the negotiation process consists of a number of decision-making episodes, each of which is characterized by evaluating an offer, determining strategies and generating a counteroffer, as illustrated in Figure 3. This negotiation process can be

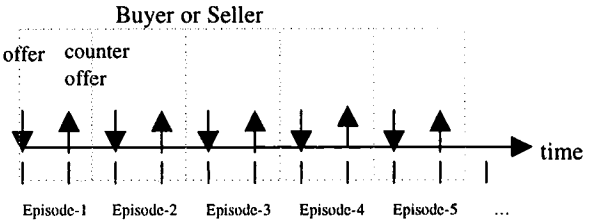

Figure 3 Negotiation process

seen as a sequential decision making model studied by Cyert & DeGroot [Cyert et. al. 87] and Bertsekas [Bertsekas 95] and used in the Bazaar model by Zeng [Zeng et. al. 98].

Within this sequential decision making paradigm, in order to acquire and learn strategy from previous experiences, it is essential to formulate a negotiation strategy based on offers/counter-offers. In each decision-making episode of negotiation, an offer/counter-offer is generated based upon an episode strategy. The negotiation strategy of overall negotiation process is a composite of all these episode strategies. The negotiation agent can change its episode strategy from episode to episode due to changing world information. However, information related to strategies is hidden from each other. In the previous negotiation experiences, a series of offers/counter-offers reflects information related to episode strategies. The changes from one offer/counter-offer to another offer/counter-offer show variation of episode strategies.

In our view, negotiation agents use episode strategies to incrementally modify offer/counter-offer parts toward an agreement. We propose using concession between offers/counter-offers to reflect episode strategies [Zhang et. al. 99]. Given a series of offers, (O1, O2, ..., O5), the concession C(i+1) applied in the episode strategy S(i+1) is formulated as a percentage based on O(i) and O(i+1)..

$$C(i+1) = [O(i+1) - O(i)] / O(i) * 100\%;$$

In this way, a series of concessions can be used to capture the implicit episode strategies used in the previous negotiation. The concession, as a percentage tries to represent context-independent information on episode strategies, which facilitates reuse of previous strategies in similar negotiation contexts. Based on this formulation, our negotiation strategy module is developed, in which previous negotiation strategies are described based upon concessions/counter-concessions of offers/counter-offers made in the negotiation. The concessions applied in

previous episodes are reused when a similar negotiation situation occurs.

Experience Based Negotiation Framework

Given an offer by opponent agent, the negotiation engine will evaluate it to decide if the offer is acceptable. If the offer is not acceptable, the negotiation engine needs to determine what episode strategy to follow in the process of generating a counteroffer. Our experience based negotiation module suggests a concession that can be used in the generation of a counter-offer (fig. 4).

Case-Based Reasoning techniques are applied in the experience-based negotiation module to represent and reuse previous negotiation experiences. The negotiation module uses strategy information from decision-making episode on previous negotiation experience to propose strategies that can be followed in a decision-making episode in the current negotiation.

The negotiation module first retrieves relevant previous negotiation experience. It then matches/selects a most matched case and finally reuses the strategy in the selected negotiation experience case(fig. 4). A number of previous negotiation cases is stored and represented in a case memory, which provide agent's profile, used car profile and relevant information to strategies used in previous negotiations. A hierarchy is used as an organizational structure for storing cases, which enable an efficient searching through cases [Zhang et. al. 99].

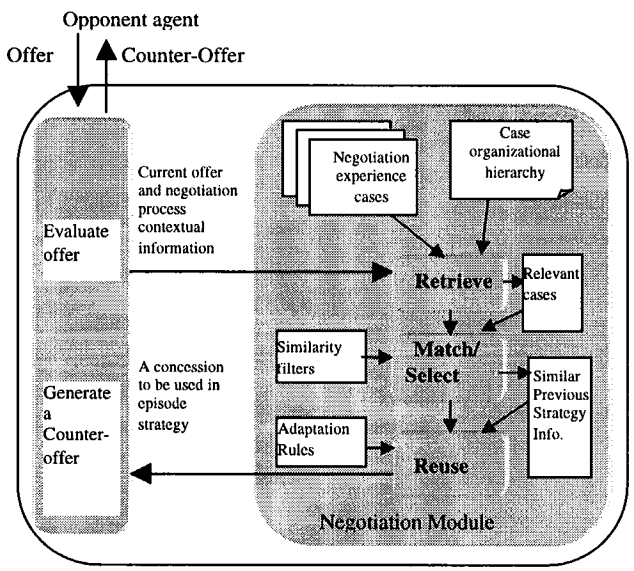

Fig 4 Our Experience Based Negotiation framework

The proposed approach is currently developed based upon the following assumptions:

- The negotiating agents are rational, i.e. the negotiation process is strictly monotonic, either decreasing or increasing depending if buying or selling.
- Single issue, only 'price' is considered in the negotiation
- Negotiation Case Base will be populated with valid and representative cases.
- All previous negotiation cases provide successful negotiation experiences. Learning from failure is not considered at this stage.

Representation of Negotiation Experiences

Previous negotiation experiences are represented as negotiation cases in our negotiation module A negotiation case represents information related to a specific agent (eg., seller or buyer) in a previous negotiation. The negotiation cases capture contextual information and negotiation experience available to an agent. A negotiation case thus contains

- Buyer's profile,
- Seller's profile,
- Used car's profile,
- Offers made from other agent and concessions used in episode strategies,
- Counter-offers made by the agent and concessions used in episode strategies,
- Performance information about the feedback of negotiation results.

The negotiation context in the module is defined by profiles of buyer, seller and used car, which provides the following information:

1) Seller's and buyer's profiles including name, age, buyer/seller's negotiation focus, expected negotiation duration, issues of negotiation (e.g. price, warranty, trade-in, etc), constraints (e.g., budget) and preferences.
2) Profile of negotiated used-car including mode, make, year.

The negotiation experience in cases can be captured by the received offers, the generated counter-offers and concessions used. The performance information gives description of outcomes of negotiations such as success/failure and final mutually agreed negotiated price.

The following is an example of a buyer agent's case.

Buyer's Profile		Seller's Profile	Profile of used-car	
Episode No	Seller's offer	Seller's concession	Buyer's counter-offer	Buyer's concession
1	O1		Co1	
2	O2	S%1	Co2	B%1
3	O3	S%2	Co3	B%2
4	O4	S%3	Co4	B%3
...	...	...	...	...
Performance information				

Figure 5: Agent negotiation case. Please note that the opponent agent's profile may be incomplete

We have presented below (fig. 6) a relational view of the case base with the case number as primary key linking all

tables-- agent profile, agent's list of offers and concessions for a negotiation case, opponent agent profile, opponent agent's list of offers and concessions for the same negotiation case, used car profile, and summary of case performance:

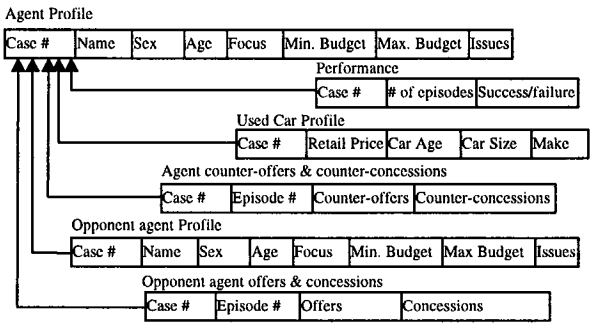

Figure 6: Relational view of the negotiation case base

Process of Experience Based Negotiation

During each negotiation episode, the negotiation module will have the information of the current negotiating agents' profiles, current car profile, and current most up-to-date episode-history-lists of offers & corresponding concessions and counter-offers & corresponding counter-concessions. Based on the current profiles of the agents and car, the negotiation module will first retrieve a relevant set of previous negotiation cases using the contextual case organization hierarchy. It will then perform similarity assessment to match/select the most similar cases from this group of relevant cases using information from all profiles and the current episode-history lists of offers, concessions, counter-offers, & counter-concessions. Information from the most similar case will be re-used to provide the next concession used to generate the next offer (fig. 2). This section will describe briefly the retrieval process and will focus on the matching & selection using the similarity measures and the reuse of the best-matched case(s) from the view point of a buyer agent.

Retrieval of negotiation case

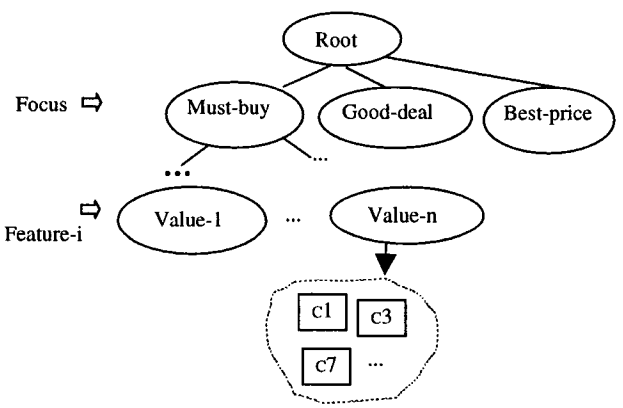

Figure 7: Case organization hierarchy for buyer agent

A contextual case organization hierarchy (fig. 7) is used as an organization structure for grouping/categorizing the negotiation cases. The context information in cases is used to classify cases in the hierarchy. A number of salient features are selected to index cases based on their possible values. For example, because we believe that different negotiation behavior arises primarily from the negotiation focus of the buying agent, the feature "focus" in the buyer agent's profile are used as primary index for different groups of buyer negotiation cases. The possible values of "focus" include "must-buy", "good-deal" and "best-price". The feature "focus" thus groups cases into three categories. Other salient features used for sub-categorization include the age and the engine-size of the car. The case organization hierarchy is used to retrieve relevant negotiation cases by searching through case memory.

If no case is found based on the case organization hierarchy, the buyer agent can fall back using some default negotiation strategies selected by the user from a set of pre-defined strategies.

Matching/Selecting similar negotiation case

Similarity/Matching filters are used to select/filter out the best-matched case from the retrieved relevant set of cases. The filters applies for all retrieved sets of relevant cases with some contextual distinction depending on the Case Organization hierarchy of "best-price", "good-deal", and "must-buy". We illustrate here the "good-deal" scenario and will show briefly the difference of "best-price" and "must-buy" scenarios.

In the "good-deal" scenario, all the retrieved relevant cases are passed first through a concession-match filter. The concession-match tries to find "sub-string" matches between the concessions & counter-concessions of the previous negotiation cases and the concessions & counter-concessions of the current negotiation process(fig 8). Dependent on the number of the filtered cases, they are then passed through an ordered series of profile matching filters. This series of profile matching filters include the starting-offer-match filter, starting-position-match filter, profile-age-match filter, and profile-sex-match filter(fig. 9).

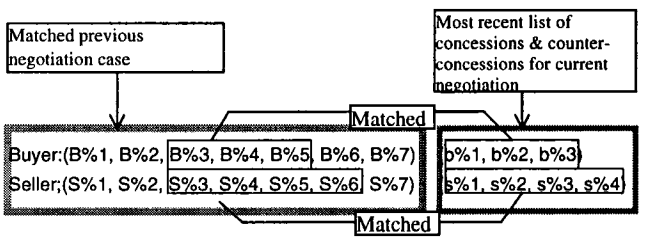

Figure 8 Concession matching.

As described earlier, Negotiation strategy for a case is reflected by the overall composite of the episodic concessions/counter-concessions during the negotiation process. The concession-match filter thus captures the negotiation processes/cases with the same strategies or sub-strategies, i.e. they are of the similar functional/sub-

functional and behavior/sub-behavior forms. The starting-offer gives preference to cases with similar initial buying offer. This filter thus restricts the magnitude difference of those cases with similar strategies or sub-strategies, i.e. this filter gives preference to negotiation cases/sub-cases which are similar in offers/counter-offers – a stronger requirement. The starting-position match filter gives preference to the previous case whose concession matches from the beginning. This filter thus restricts the translational difference. It throws away the sub-strategy functional match and sub-case match and prefers full-strategy/ full-case match. The age and sex filters respectively match cases with the same age range and sex. They are used to put some preference on the matching cases if the two restrictive (starting-offer-match and starting-position-match) do not produce any matching candidates. Because of the noisy nature of the cases, those filters are arranged in such manner to give most opportunity of getting some best-matched case(s).

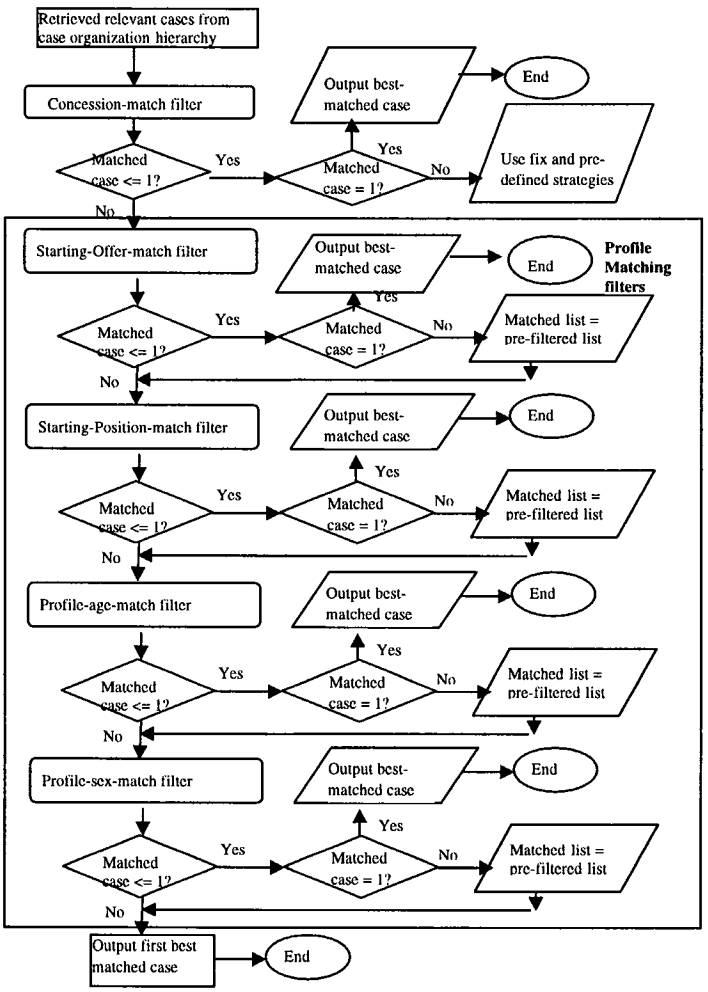

Figure 9: Similarity assessment for good-deal scenario

"Best-price" and "Must-buy" scenarios have case similarity/matching assessment methodologies of the same nature. For example, in the "Must buy" scenario, there is an addition rule before the concession-match filter and profile match filters which says:

If the opponent agent's offer reaches a plateau and offer is within the agent's budget, suggest a concession that can reach the opponent agent's offer.

Reuse of Previous Concessions

Once a case is selected as the best matched case to the current scenario, the negotiation module takes the next concession that was used in previous negotiation and suggests it to the negotiation agent for the generation of a counter-offer. For example, in figure 6, the strategy "B%-6" will be reused to recommend to the negotiation agent to follow for generating a counter-offer.

Some adaptation is done during the boundary conditions, e.g. if the concession results in a counter-offer which is higher than the buyer's maximum budget, then the concession must be adjusted accordingly to arrive only at the counter-offer equal to or not greater than the buyer's maximum budget.

Systems design and implementation

Our Experience Based Negotiator prototype is designed as a distributed three-tier server client architecture: Negotiator client, Negotiator server, and Negotiation Case Repository(fig. 10). Because of the benefits from portability, object-oriented design, multi-threading, flexibility and web advantage, Java 1.2 is the programming language used for implementation.

The Negotiation Case Repository manages the Case Base, i.e. all the stored negotiation cases. The Case Repository is implemented via a standard SQL capable relational database management system. All the cases are captured within several relational tables in a database.

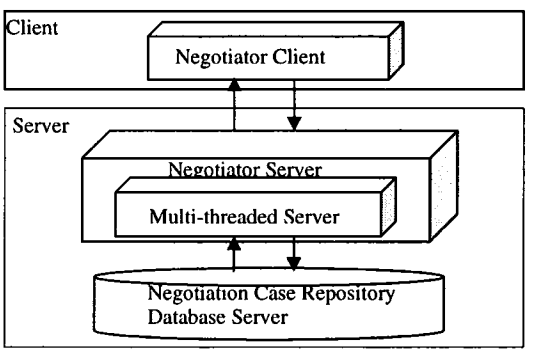

Figure 10. Client-Server architecture

The Negotiator server stands in between the Negotiator client and the Negotiation Case Repository. Using specific communication protocol via socket connection, it accepts input from the Negotiator client and generates output to the client. It also accesses the Negotiation Case Base directly using SQL queries via Java Database Connectivity(JDBC) API. The Negotiator server is the reasoning engine which applies various similarity assessments to find the best

matched case from a pool of relevant cases retrieved from the Case Base and adapts best-matched case to obtain episodic strategy to generate counter-offer. It is also the data processing engine for case browsing and descriptive statistical reports.

The Negotiator server is actually multi-threaded to allow for several clients to connect simultaneously. All clients initiate connection with the negotiator server via an advertised listening port. Once the initiating connection is made, the server will assign a new dedicated port for future continual communication between the server and client. The advertised listening port is released for other new clients.

The negotiator client provides the user interface and socket connection to server via internal communication protocol. The graphical user interface provides data collection, data representation and display facilities. The graphical user interface is implemented using the Swing API. When designing this server-client system, we have emphasized lessening processing load on the client-side. This is to compensate for the extensive processing the client has to do to maintain the interactive user interface.

Discussion

To deal with the changing world information, the negotiation agent needs to revise and adapt its negotiation strategies from session to session during negotiation. The negotiation strategy needs to be defined based upon the knowledge, past experience and available information. This paper has described an on-going research project for providing adaptive negotiation strategies in negotiation. The proposed approach is characterized by its capability of learning from experiences, which can help the negotiation process improve its performance by providing adaptive contextual appropriate strategies. The current module is developed for single-issue (e.g., car-price) negotiation. An assumption is made that all cases provide successful negotiation experiences. Learning from failure is not considered.

One popular approach towards solving conflict resolution in multi-agent negotiation is to make use of utility theory [Sycara, 88]. Utility theory is the theory that models the process through which a decision-maker evaluates a set of alternatives, so that he/she can choose the best one. Utility theory provides a measure of an overall utility or satisfaction of a decision-maker and as a result, indicates a way towards achieving a maximum utility or satisfaction of a decision. In general, the theory accommodates multiple issue negotiation (e.g. price, warranty, etc.). In our current scope, only a single issue, namely car-price, forms the core of our negotiation. As such, the utility functions of our selling agent U(S) and buying agent U(B) looks respectively like:

U(B)= f1(Budget - Price);U(S) = f2(Price - cost);

Clearly, depending on the various subjective criteria among the agents, for example, focus="Must-Buy", or focus="Good-Deal" or focus="Best-Price", the functional shape of f1 and f2 varies significantly among different selling agents and different buying agents.

We believe strongly that one effective way to capture these utility functions is to examine the previous behavior of the agents. In our case, we look at all successful negotiation experiences of the buyer and seller agents and assume those collective experiences indicate well the inherent utility functions of a particular type of agent under a particular type of circumstances. We then adapt and apply the result to current negotiation situation. In doing so, we avoid explicitly specifying the utility functions by assigning arbitrary values. Moreover, our case-based/ experience-based approach can allow us to accommodate easily the change/growth of the utility functions e.g., through time [Zhang et. al. 99].

Acknowledgements

We are very much in debt to Dr Ryszard Kowalczyk , Mr. Van Bui and other CSIRO AAI research group for their invaluable inputs and discussion.

Reference

[Bertsekas 95] Bertsekas, D. P., *Dynamic Programming and Optimal Contral.* Athena Scientific, Belmont, MA 1995.

[Chavez et .al. 97] Chavez, A.,Dreilinger, D., Guttman, R., and Maes, P. *A real-life experiment in creating an agent marketplace.* Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM'97.

[Cyert et .al. 87] Cyert, R. M. and DeGroot, M. H., *Bayesian Analysis and Uncertainty in Economic Theory,* . Rowman & Littlefield, New York, 1987.

[Kowalczyk R et .al. 99] Kowalczyk R. and Bui V. *Towards Intelligent Trading Agents.* The International Conference on Intelligent Systems and Active DSS in Turku/Abo, Finland, 1999

[Maes et .al. 99] Maes, P., Guttman, R. H., and Moukas, A. G. *Agents that buy and sell.* Communications of The ACM. Vol. 42. No.3 pp. 81-91, 1999.

[Sandholm 99] Sandholm, T, *Automated Negotiation: The best terms for all concerned* Communications of The ACM. Vol. 42. No.3 pp. 84-85, 1999.

[Siriwan et. al. 99] Siriwan, A and Sadananda R., *An agent-mediated Negotiation Model in Electronic Commerce,* The Australian Workshop on AI in Electronic Commerce, The Australian Joint Conference on Artificial Intelligence 1999.

[Sycara 88] Sycara, K. *Utility theory in conflict resolution.* Annals of Operations research, Vol. 12 pp. 65-84, 1988.

[Sycara 93] Sycara, K. *Machine learning for intelligent support of conflict resolution,* Decision Support systems, Vol. 10, pp. 121-136, 1993.

[Zeng et. al. 98] Zeng, D and Sycara, K. *Bayesian Learning in Negotiation,* Int. J. Human-Computer Studies 48, pp 125-141, 1998.

[Zhang et. al. 99] Zhang, D., Wong, W., and Kowalczyk, R. *Reusing Previous Negotiation Experiences in Multi-Agent Negotiation,* Proceedings of WAEC Workshop, 1999.