An Implementation of Collective Collaboration Support System based on Automated Multi-Agent Negotiation

著者（英）	Mikoto Okumura, Katsuhide Fujita, Takayuki Ito
journal or publication title	Studies in Computational Intelligence
volume	435
page range	125-141
year	2013
URL	http://id.nii.ac.jp/1476/00003659/

doi: 10.1007/978-3-642-30737-9-8(http://dx.doi.org/10.1007/978-3-642-30737-9-8)

Implementation of Collective Collaboration Support System based on Automated Multi-Agent Negotiation

Mikoto Okumura
Nagoya Institute of Technology
Gokiso-cho, Nagoya-shi, Aichi, 466-8555 Japan
okumura@itolab.mta.nitech.ac.jp

Katsuhide Fujita
Nagoya Institute of Technology
Gokiso-cho, Nagoya-shi, Aichi, 466-8555 Japan
fujita@itolab.mta.nitech.ac.jp

Takayuki Ito
Nagoya Institute of Technology
Gokiso-cho, Nagoya-shi, Aichi, 466-8555 Japan
ito.takayuki@nitech.ac.jp

ABSTRACT

Recently, discussions among many people about global warming and global product development have been increasing. Efficient collaborative support based on multi-agent systems is necessary to collect the huge number of opinions and reach optimal agreements among many participants. In this paper, we propose a collaborative park-design support system as an example of collective collaboration support systems based on multi-agent systems. In this system, agents elicit the utility information of users, collect many alternatives, and reach optimal agreements based on automated negotiation protocol. In particular, we focus on the steps for determining the attribute space and estimating the utility spaces of users in real world.

1. INTRODUCTION

Recently, discussions among many people about global warming and global product development are increasing. Efficient collaborative support based on multi-agent systems is necessary to collect huge number of opinions and reach optimal agreements among many participants. Many automated negotiation mechanisms are existed, however, the perfect utility functions of agents are assumed [1, 2, 3, 4, 5, 6, 7, 8]. In real world, it takes a lot of time to elicit the whole utility spaces of users.

In this paper, we propose a collective collaboration support system based on the multi-issue automated negotiation mechanisms [1, 2, 3, 4, 5, 6, 7, 8]. In this system, the agents elicit the utility information of users, collect many alternatives, and reach optimal agreements based on the automated negotiation protocol. Especially, we focus on the steps of deciding the attribute space and estimating the utility spaces of users in real world.

In this paper, we adopt a collaborative park-design support system as an example of a collective collaboration support system. Many users, like citizens and designers, should join the work to design parks. Many opinions and preferences of participants should be respected. Additionally, the designs of parks have some interdependent issue, for example, there are some dependence between the amount of playground equipments and the cost. In such a case, the automated negotiation protocol with issue-interdependency is effective [5, 6, 7, 8]. However, to apply the automated negotiation protocol with issue-interdependency, we need utility functions of users because most of the papers assumed the perfect utility functions of agents. In real world, it is impossible to elicit all the utility information of agents.

Our system estimates the interdependent multi-attributes utility functions of users based on users' evaluation of the designs generated by our system. In this paper, the utility function is composed of some simple fundamental functions. One fundamental function is defined by one user's evaluation of designs. The fundamental functions has a character that the utility grows low as the point is far from the sampling point corresponding to the design. The bumpy utility space is generated by combining the simple mound of the fundamental functions.

The remainder of this paper is organized as follows. First, we describe the outline of the collaborative park-design support systems. Next, we propose a new method of estimating the utility functions of users in real life. Third, we demonstrate some results of our method and conduct an experiment to evaluate the effectiveness of our system. Finally, summarizing our paper.

2. COLLABORATIVE PARK DESIGN SUPPORT SYSTEM

Figure 1 shows the outline of a collaborative system based on multi-agent system. The details of the system are followings.

`[Step1]` **Collecting the opinions and preferences**
The system decides sampling points and generates the alternative of the park design at the sampling point. After that, this system elicit the uses' preferences based on users' evaluations.

`[Step2]` **Estimating utility functions**
The system predicts the whole utility spaces based on the sampling points collected in the `[Step1]`. This estimation will be shown in the section3.

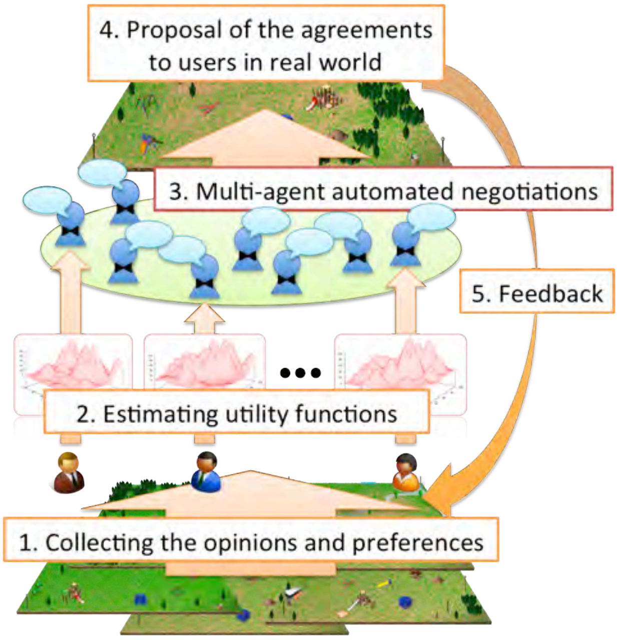

Figure 1: Collaborative public space design processes

[Step3] **Multi-agent automated negotiations**
The agent which is behalf of the users finds the optimal agreements by the automated negotiation protocol[1, 2, 3, 4, 5, 6, 7, 8].

[Step4] **Proposal of the agreements to users in real world**
The system generates a design from the agreement (a point on the attribute space) in [Step3].

[Step5] **Feedback**
The system sends the design result(final alternative) in this round, and the users give a feedback. If the most of users agree to the final alternative, it is an optimal agreement. If the most of users don't agree to the final alternative, these steps are repeated([Step1]∼[Step5]). But, our system doesn't have this step to be simple because our study and implementations of system based on multi-agent system are in its early stage.

3. METHOD OF ELICITING THE UTILITY SPACES

The method of eliciting utility functions corresponds to the [Step1], [Step2] in the section2. This system generates the park designs automatically, receives the users' evaluations, and estimates the utility spaces.

We employ the parametric design because the parameters in the attribute space correspond one-to-one with park designs. In other words, our system can convert an agreement point in the automated negotiation to the park-design in real world.

In this paper, the utility function is composed of some simple fundamental functions. The fundamental functions has a character that the utility grows low as the point is far from the sampling point. The bumpy utility space is generated by combining the simple mound of the fundamental functions.

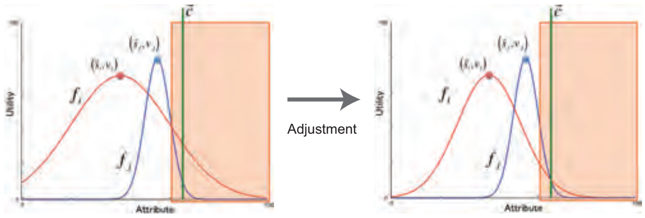

Figure 2: The Fundamental Function is Under the Other Fundamental Function (Case1)

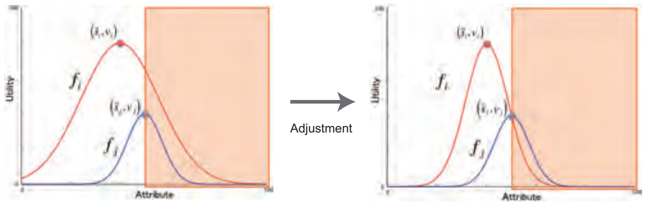

Figure 3: Most of the Fundamental Function are Under the Other Fundamental Function (Case2)

3.1 Fundamental Function

DEFINITION 1. *Fundamental Function*

$\mathbb{R}^+$ is a set of positive real numbers more than 0, $\mathbb{R}^*_+$ is a set of all positive real numbers. When i is an index, s_i shows a sampling point, d_i is the distribution of f_i and v_i is the evaluation value of $s_i(v_i, d_i \in \mathbb{R}^*_+)$. The fundamental function f_i is defined as a following expression.

$$f_i(\vec{x}) = v_i \cdot \exp(-\frac{(\vec{x} - \vec{s_i})^2}{d_i}) \quad (1)$$

- The fundamental function is always more than 0 and a multi-dimensional space.

$$f_i : \mathbb{R}^{+n} \to \mathbb{R}^+$$

- The maximum of the fundamental function is equal to the evaluation value of the user.

$$\max f_i(\vec{x}) = v_i$$

- The maximum point of fundamental function means the sampling point.

$$\arg\max_{\vec{x}} f_i(\vec{x}) = \vec{s_i}$$

- The value of the fundamental function is smaller as it grows far from the sampling point.

$$||\vec{x_1} - \vec{s_i}|| > ||\vec{x_2} - \vec{s_i}|| \to f_i(\vec{x_1}) < f_i(\vec{x_2})$$

3.2 The combination of the fundamental functions

DEFINITION 2. *Utility Function*

When there are N sampling points; $(\vec{s_1}, .., \vec{s_N})$, the utility function U is defined as follows:

$$U(\vec{x}) = \max_{i=1,..,N} f_i(\vec{x}) \quad (2)$$

However, the definition 2 has two main problems as Figure 2, Figure 3 shows. In the left part of Figure 2, the sampling s_j don't work well because the function f_j is totally smaller than the function f_i. For instance, our system should employ f_j at the square area because the sample point of f_j (s_j) is closer to the square area than that of f_i (s_i). In the left part of Figure 3, the sampling s_j don't work well because most of the function f_j is smaller than the function f_i. For instance, our system should employ the function f_j at the square area in the Figure 2 because the point in the square area is closer to the sampling point of function f_j (s_j) than that of function f_i (s_i). Following two techniques resolve these two problems by modifying d_i which is the distribution of the fundamental function f_i.

METHOD 1. *A fundamental function is under other fundamental function (Case 1)*

This method adjusts the f_i as Figure 2 showing by modifying d_i. For instance, we assume that two different sampling points $\vec{s_i}, \vec{s_j}((i \neq j, \max f_i(\vec{x}) \geq \max f_j(\vec{x}))$ exist. If $f_i(\vec{s_j}) > f_j(\vec{s_j})$, then this method modifies d_i to satisfy $f_i(\vec{s_j}) = f_j(\vec{s_j})$ using the expression(3).

$$d_i = \frac{(\vec{s_j} - \vec{s_i})^2}{\ln \frac{v_i}{v_j}} \quad (3)$$

METHOD 2. *Most of a fundamental function are under other fundamental function (Case2)*

This method adjusts the f_i as Figure 3 showing by modifying d_i. For instance, we assume that two different sampling points $\vec{s_i}, \vec{s_j}(i \neq j, d_i > d_j)$ exist. If $f_i(\vec{c}) > f_j(\vec{c})$, then this method modifies d_i to satisfy $f_i(\vec{c}) = f_j(\vec{c})$ by the expression(4).

$$d_i = \frac{(\vec{c} - \vec{s_i})^2}{\frac{k^2}{2} - \ln \frac{v_j}{v_i}} \quad (4)$$

$\vec{c} = \vec{s_j} + k\sqrt{\frac{d_j}{2}}\vec{u}$, $\vec{u} = \frac{1}{||\vec{s_j} - \vec{s_j}||}(\vec{s_j} - \vec{s_j})$. $\vec{u}$ is the unit vector whose direction is from $\vec{s_i}$ to $\vec{s_j}$. $\vec{c}$ is a control point of adjusting. $\vec{c}$ depends on the parameter $k(\in \mathbb{R}^*_+)$. As k grows, this method is performed at the point which distance from s_j is large. For example, $\vec{c}$ goes right as k grows in Figure 3.

The simple way of estimating utility functions is to connect all sampling points smoothly as Figure 4 showing. However, the utility is usually estimated as higher value than real one when the distance between some sampling points is large. Our method improves this issue by using a maximum of fundamental functions as Figure 4 showing.

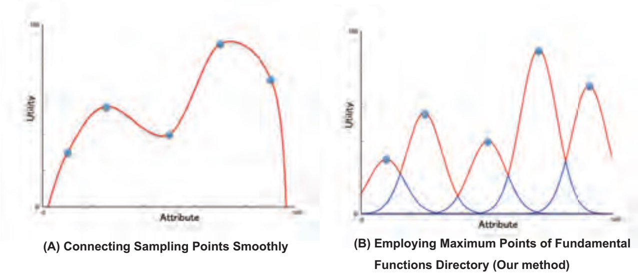

Figure 4: A Estimated method of Connecting Sampling Points Smoothly and Employing Maximum Points of Fundamental Functions Directory

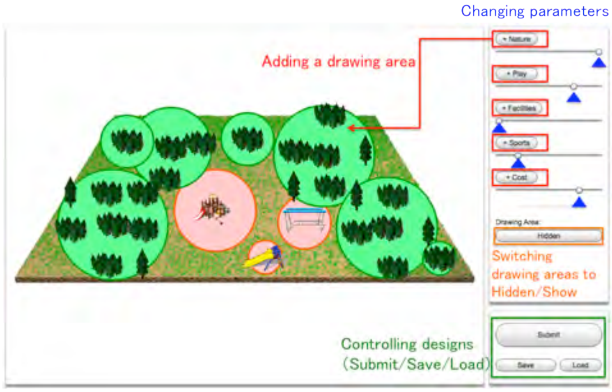

Figure 5: User Interface of Creating a Fundamental Park Design

Our method has a tendency to make agreement at the area of containing more information because the utility with enough information is large. By contract, it is difficult for agents to make agreement when the number of samples is not enough. However, our method modifies this problem because the sampling points are decided based on the users' suggestions.

3.3 Estimating the Utility Space of users in real world

In this paper, our system estimates the user's utility space as follows(`[Step1]`〜`[Step4]`).

`[Step1]`**Creating a Fundamental Design**
The manager of the negotiation sets up a negotiation. The manager creates a fundamental design by the user-interface as Figure 5 showing. The manager decides the arrangements of trees, playground equipments, facilities and so on. The manager can check the park designs generated automatically by our system, and change some parameters for reflecting his ideas.

`[Step2]`**Deciding Sampling Points**
Our system decides some sampling points in the attribute space. In this paper, the sampling points are selected randomly.

`[Step3]`**Evaluation by the users**
Our system generates the park designs at the sampling points. There are some appraising methods for evaluating the sam-

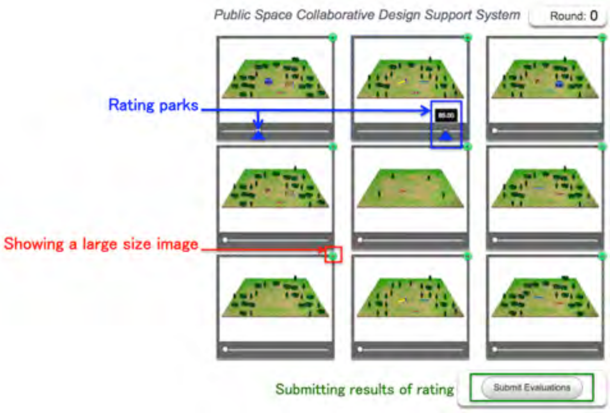

Figure 6: User Interface of Evaluating Park Designs

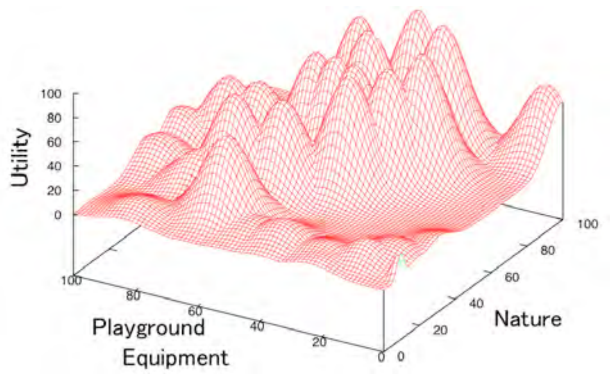

Figure 7: Estimated Utility Function

pling points (e.g. voting, rating). In this paper, we employ the rating method. Users rate each park design and submit the results of rating by the user-interface as Figure 6 showing.

`[Step4]` **Estimating Utility Functions**
First, the system generates the fundamental functions. Next, our system combines all of the fundamental functions by Method1 and Method2). Specifically, these methods adjust $d_i, d_j (0 \leq j < i)$. d_i is initialized by $D_0 (d_i := D_0$, D_0 is the initial value of the distribution of fundamental function).

`[Step2]`∼`[Step4]` is repeated during the period of negotiation decided by the manager in the [Step1].

4. AN EXAMPLE OF ESTIMATING UTILITY FUNCTION

In this section, we demonstrate some results of our systems. The purpose of the demonstration is to evaluate our method and to show the characters of our method.

In this demonstration, we assume that the user has a following idea: "The parks which have many trees and some playground equipments are good. The parks which have too many or few playground equipments are not so good." Figure 7 shows an example of the estimated utility function. In this demonstration, the number of sampling is 30 and the number of attributes is 2. The reason of small number of attributes is that we can't show graphically when the number of attributes is more than 3. As you know, our method can be applied when the number of attributes is more than 3.

The axis "Nature" in Figure 7 shows how rich the nature of the park is. The large value of "Nature" means that the park has rich nature. The axis "Playground Equipment" shows how many the playground equipments are in the park. The large value of "Playground Equipment" means that the park has many playground equipments.

In Figure 7, the utility is the highest when "Playground Equipment" is 50 and "Nature" is the high value. Therefore, the estimated utility function represents the accurate preferences of the user. However, the utility is too high when "Nature" is 60 and "Playground Equipment" is 50. This is because that too many samplings are happened at the point. The efficient sampling for estimation of utility spaces is one of the future work.

	U1	U2	Est1	Est2	Err1	Err2
User1	80	85	94	61	14	24
User2	80	80	-	64	-	16
User3	60	50	19	74	41	24
User4	80	85	-	-	-	-
User5	85	75	80	73	5	2
User6	90	97	62	82	28	15
User7	70	80	57	51	13	29
User8	90	86	67	58	23	28
User9	80	80	90	87	10	7
User10	90	90	86	75	4	15
User11	65	65	69	71	4	6
Average	79.09	79.36	69.30	69.60	15.78	16.60

Table 1: Utility Values for The Optimal Agreement

5. EXPERIMENTS

5.1 Setting of Experiments

We conducted an experiment to evaluate the effectiveness of our system and confidence of our preference elicitation method. In the experiment, we ran 2 negotiations. In each negotiation, a number of participants is 11. "Nature" and "Playground Equipment" are used as attributes. Each attribute is a real number which is bigger than or equal to 0 and less than or equal to 100. The period of first negotiation is 10 minutes and second negotiation is 5 minutes. Because participants understood our system, the period of second negotiation is reduced. To find the optimal agreement, we used simulated annealing (SA) and the best result of 5 SAs is adopted as the optimal agreement because SA is easy to implement and finding optimal agreement is not our main work. After the negotiations, we send out questionnaires to get users' comments.

5.2 Results of Experiments

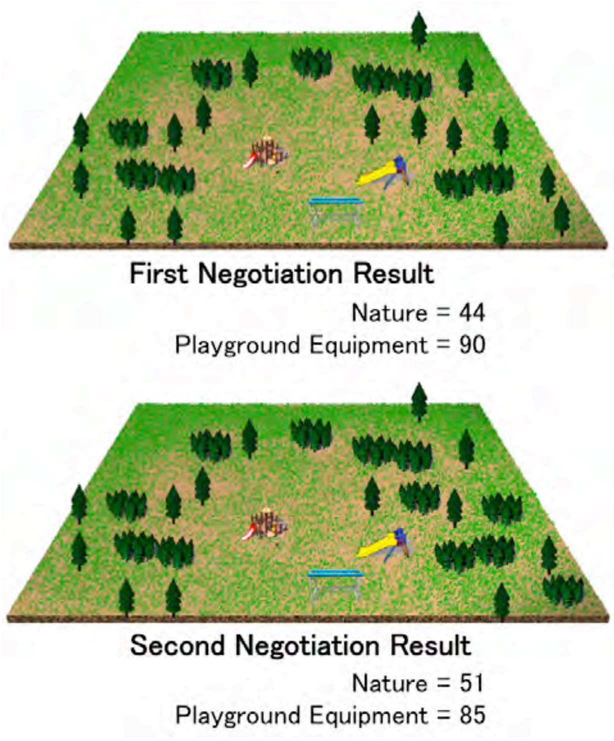

Figure 8: Negotiation Results

Figure 8 shows the results of negotiations. Table 1 shows some information about utility values for the results. U1 and U2 are user's rates (real utility values) for the results. Est1 and Est2 are user's estimated utility values for the results. Err1 and Err2 are margins of error between real utility value (U1, U2) and estimated utility value (Est1, Est2). U1, Est1 and Err1 are values for the first negotiation. U2, Est2 and Err2 are values for the second negotiation.

An average of U1 is 79.09 and U2 is 79.36. Therefore, we find many people agree to the results. An average of Err1 is 15.78 and Err2 is 16.60. These values are not so good but our method of preference elicitation can elicit tendencies of user's preference.

Table 1 shows that most of the users' utility function are accurately elicited like User5 but some users' preference elicitation are not accurately elicited like User3. Figure 9 is a elicited utility function of User5. This case is preference of a user is accurately elicited. Figure 10 is a elicited utility function of User3. This case is preference of a user is not accurately elicited. A shape which has many sharp mounds like Figure 10 occurs when some close points on a attributes space have very different utility values, in other words, some similar park designs are got very different rates. A reason of this problem is no considering changes of preferences. In fact, User3 commented "My criteria of rating are inconsistent in a process of evaluations." on a questionnaire. In real world, human preferences change as time passes. Additionally, almost all methods of preference elicitation[9, 10, 11]

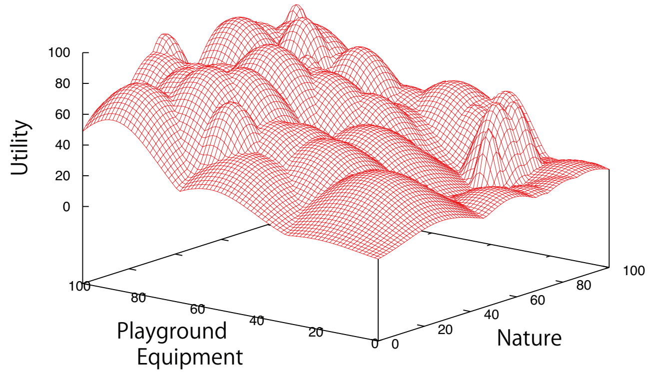

Figure 9: A Good Case of Elicitation (Elicited Utility Function of User5)

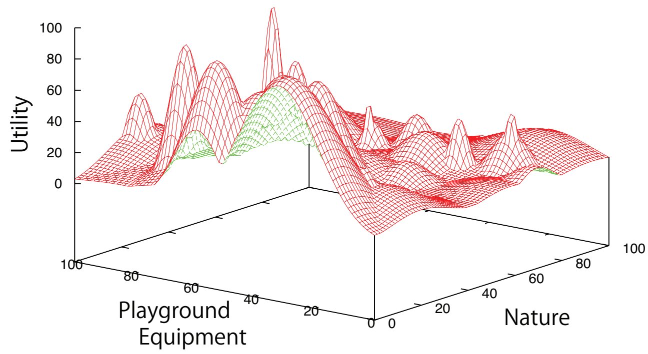

Figure 10: A Bad Case of Elicitation (Elicited Utility Function of User3)

and utility theory has this problem, too. It seems establishing "time" as a attribute resolves this problem but sampling points on a utility space from a user can not be finished. Because a number of samples getting at once is limited and samples acquired on different steps have different time as an attribute. As a result, a number of samples can not be enough to describe a utility space.

6. RELATED WORKS

Most previous works on multi-issue negotiation have addressed only linear utilities[1, 2, 3, 4]. Recently some researchers have been focusing on more complex and nonlinear utilities. For example, Ito, Fujita and Mizutani et. al[5, 6, 7, 8] proposes the automated negotiation protocol with issue-interdependency. However, most of the paper assumed the perfect utility functions of agents. In real world, it is impossible to elicit the all utility information of agents. In this paper, we propose the method of estimating the utility spaces with issue-dependences.

Luo et al.[9] proposes a method of eliciting and quantifying the trade-off between issues by the user-interactions. However, the system don't work well when the utility function is complex with the dependences between more than 3 issues. On the other hand, our system can work well when the utility functions are more complex.

In [12, 13], the system supports a yard-design based on interactive GA. This system can generate the efficient yard-designs based on the preference of users. However, this system isn't assumed the multi-party negotiations. Our system supports to the multi-party collaborative designs and consensuses among many users.

7. CONCLUSION

In this paper, we implemented a collaborative park-design support system based on the multi-agent systems. In particular, we focused on the steps for determing the attribute space and estimating the utility spaces of users in real world. Our experimental results shows our system succeeded to build a consensus many participants agreed to.

Our future works are the method of selecting sampling points for efficient estimation of utility spaces, an implementation of a feedback step and establishing a new model considerd changes of human preferences.

8. REFERENCES

[1] Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. In Artificial Intelligence, pages 142:205-237, 2002.

[2] S. Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. Multi-issue negotiation with deadlines. J. Artif. Intell. Res. (JAIR), 27:381-417, 2006.

[3] S. Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. Approximate and online multi-issue negotiation. In AAMAS, page 156, 2007.

[4] S. Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions. In AAMAS (2), pages 1041-1048, 2009.

[5] Ito, T., Hattori, H., and Klein, M.: Multi-issue Negotiation Protocol for Agents : Exploring Nonlinear Utility Spaces, in Proc. of IJCAI-2007, pp. 1347?1352 (2007).

[6] Katsuhide Fujita, Takayuki Ito, Mark Klein: Approximately fair and secure protocols for multiple interdependent issues negotiation. AAMAS (2) 2009: 1287-1288.

[7] Katsuhide Fujita, Takayuki Ito, Mark Klein: Secure and efficient protocols for multiple interdependent issues negotiation. Journal of Intelligent and Fuzzy Systems 21(3): 175-185 (2010).

[8] Nobuyasu Mizutani, Katsuhide Fujita, Takayuki Ito: Issue Clustering and Distributed Genetic Algorithms for Multi-issue Negotiations. ACIS-ICIS 2010: 593-598.

[9] X Luo, N. R. Jennings, and N. Shadbolt. Acquiring user strategies and preferences for negotiating agents: A default then adjust method. International Journal of Human Computer Studies, 64(4):304-321, 2006.

[10] Kuriyama Koichi and Ishii Yutaka. Estimation of the Environmental Value of Recycled Wood Wastes : A Conjoint Analysis Study. Journal of forest research 5(1), 1-6, 2000-02-16.

[11] Tuomas Sandholm and Craig Boutilier. Preference Elicitation in Combinatorial Auctions. Proceedings of the Third ACM Conference on Electronic Commerce, 2001.

[12] Nanba Masayoshi, Kobe Masashi and Cai Dongsheng. Japanese Garden Design using Interactive GA. In IPSJ SIG Notes of Information Processing Society of Japan (IPSJ): 2001(96), 55-62, 2001-10-12.

[13] Nanba Masayoshi, Kobe Masashi and Cai Dongsheng. Designing Japanese Garden using Interactive GA. In IPSJ SIG Notes of Information Processing Society of Japan (IPSJ): 2003(86), 53-58, 2003-08-18.

APPENDIX

A. THE PRODUCTION OF THE EXPRESSION (3)

The expression (3) modifies d_i to satisfy $f_i(\vec{s_j}) = f_j(\vec{s_j})$.

$$\begin{aligned} f_i(\vec{s_j}) &= f_j(\vec{s_j}) \\ v_i \cdot \exp(-\frac{(\vec{s_j} - \vec{s_i})^2}{d_i}) &= v_j \\ d_i &= \frac{(\vec{s_j} - \vec{s_i})^2}{\ln \frac{v_i}{v_j}} \end{aligned}$$

d_i is produced.

To f_i becomes the gaussian, d_i must be positive.

$$\begin{aligned} d_i = \frac{(\vec{s_j} - \vec{s_i})^2}{\ln \frac{v_i}{v_j}} &> 0 \\ v_i &> v_j \end{aligned}$$

Because of $f_i(s_j) > f_j(s_j)$ which is the condition of applying the method 1,

$$v_i = f_i(s_i) > f_i(s_j) > f_j(s_j) = v_j$$

$d_i > 0$ is evidenced.

B. THE PRODUCTION OF THE EXPRESSION (4)

The expression (4) modifies d_i to satisfy $f_i(\vec{c}) = f_j(\vec{c})$.

$$\begin{aligned} f_i(\vec{c}) &= f_j(\vec{c}) \\ v_i \cdot \exp(-\frac{(\vec{c} - \vec{s_i})^2}{d_i}) &= v_j \cdot \exp(-\frac{(\vec{c} - \vec{s_j})^2}{d_j}) \end{aligned}$$

Because of $\vec{c} = \vec{s_j} + k\sqrt{\frac{d_j}{2}}\vec{u}$,

$$\exp(-\frac{(\vec{c} - \vec{s_i})^2}{d_i}) = \frac{v_j}{v_i} \cdot \exp(-\frac{k^2}{2}\vec{u}^2)$$

$\vec{u}^2 = 1$ because $\vec{u}$ is a unit vector.

$$\begin{aligned} \exp(-\frac{(\vec{c} - \vec{s_i})^2}{d_i}) &= \frac{v_j}{v_i} \cdot \exp(-\frac{k^2}{2}) \\ d_i &= \frac{(\vec{c} - \vec{s_i})^2}{\frac{k^2}{2} - \ln \frac{v_j}{v_i}} \end{aligned}$$

d_i is produced.